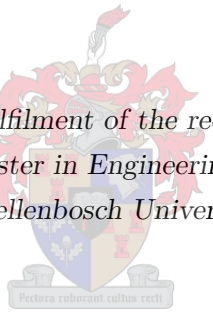
A GPS-based On-board Orbit Propagator for Low Earth-Orbiting CubeSats

by

Nico Chris Rossouw



Thesis presented in partial fulfilment of the requirements for the degree of
Master in Engineering
at Stellenbosch University

Supervisor:

Prof W.H. Steyn

Department Electrical and Electronic Engineering

December 2015

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the owner of the copyright thereof (unless to the extent explicitly otherwise stated) and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

December 2015

Abstract

On-board knowledge of satellite position is vital to all space missions. Due to the stringent power budgets of the CubeSat form factor, permanently active on-board GPS receivers for accurate navigation solutions are infeasible. Analytical techniques such as SGP4, which have been used since the 1970s, are often identified as the natural replacements. Unfortunately, due to inherent position errors of more than 1 km RMS, such techniques are unfit for some missions that prioritise precision. This emphasises the need for a new strategy for accurate on-board orbit determination and propagation.

In an attempt to conserve power, it is proposed that the GPS receiver is activated intermittently such that a duty cycle of less than 15% (including the time-to-first-fix) is required. The orbit will be estimated during the activated period, after which navigation solutions will be obtained by means of propagation algorithms during the deactivated parts. Two approaches, one based on an analytical method and the other on numerical integration, are designed and implemented in the C programming language. These systems are simulated using actual on-board GPS datasets from SumbandilaSat and NigeriaSat missions.

The analytical approach is based on SGP4. TLE parameter offsets are estimated by filtering the difference between the GPS and SGP4 output instantaneous Kepler elements, and an additional time offset is calculated for an in-track correction. Simulations revealed RMS and maximum 3D position errors of 200 m and 1 km, respectively, for a 12.5% duty cycle.

The numerical approach employs an Extended Kalman Filter with a 4^{th} order Runge-Kutta integrator for propagation. Reducing the propagator's orbit dynamics model to only a 9^{th} order JGM-3 geopotential and a Jacchia atmospheric density model for aerodynamic drag still produced fairly accurate results. Simulations revealed RMS and maximum 3D position errors of 60 m and 300 m, respectively, for a 10.7% duty cycle.

The proposed system therefore delivers a remarkable accuracy improvement over standard analytical propagators at only a fraction of the power required by permanently active on-board GPS receivers.

Uittreksel

Dit is krities belangrik dat 'n satelliet ten alle tye kennis van sy eie posisie dra. Die streng kragverbruik begroting van 'n CubeSat veroorsaak dat akkurate navigasie d.m.v. 'n deurlopend geaktiveerde aanboord GPS ontvanger nie lewensvatbaar is nie. Analitiese metodes soos SGP4, wat al vanaf die 1970s gebruik word, is gewoonlik die naasbeste oplossing. Laasgenoemde se inherente wortel gemiddelde kwadraat (WGK) posisie afskattingsfoute van meer as 1 km is onaanvaarbaar vir sekere satelliet missies. 'n Nuwe strategie vir akkurate aanboord wentelbaan afskatting en voorspelling is dus nodig.

Daar word voorgestel dat die GPS ontvanger sporadies aangeskakel word om krag te bespaar. 'n Diens-siklus van minder as 15% (met die tyd-tot-eerste-oplossing van GPS ontvanger ingesluit) word aanbeveel. Die wentelbaan sal gedurende die aangeskakelde periode afgeskat word, waarna algoritmes die posisie van die satelliet sal voorspel. Twee benaderings, een op 'n analitiese metode gebaseer en die ander een op numeriese integrasie, is ontwerp en in die C programmeringstaal geïmplementeer. Hierdie stelsels is gesimuleer met egte SumbandilaSat en NigeriaSat aanboord GPS data.

Die analitiese metode is op SGP4 gebaseer. Aanpassingsveranderlikes vir die TLE parameters is bepaal deur die verskil tussen die GPS en SGP4 uittree se oombliklike Kepler elemente te gefiltreer. 'n Tydaanpassingsveranderlike is ook bereken om 'n finale in-spoor regstelling te maak. Simulasies het WGK en maksimum 3D posisiefoute van onderskeidelik 200 m en 1 km behaal met 'n 12.5% dienssiklus.

Vir die numeriese benadering is 'n Uitgebreide Kalman Filter, met 'n 4^{de} orde Runge-Kutta integreerder vir voorspelling, ontwikkel. Die dinamika model was beperk tot 'n 9^{de} orde JGM-3 geopotensiaal en 'n Jacchia atmosferiese digtheidsmodel vir aërodinamiese sleur, en dit het steeds goeie resultate gelewer. Simulasies het WGK en maksimum 3D posisiefoute van onderskeidelik 60 m en 300 m behaal met 'n 10.7% dienssiklus.

Die voorgestelde oplossing lewer dus 'n stelsel wat 'n merkwaardige akkuraatheid verbetering teenoor gewone analitiese metodes behaal teen slegs 'n fraksie van die kragverbruik van 'n deurlopend geaktiveerde aanboord GPS ontvanger.

Acknowledgements

I would hereby like to express my sincere gratitude to the following:

- Prof W.H. Steyn, for his guidance and willingness to share his prestigious knowledge.
- My ESL colleagues, in particular Nico Calitz, Douw Steyn, Muhammad Junaid, Mohammed Bin Othman, Willem Jordaan, Mike-Alec Kearney, André Heunis, Gerhard Janse van Vuuren and Jan-Hielke le Roux, for providing invaluable advice and creating a fun working environment.
- My father, Louis Rossouw, for providing wisdom, inspiration as well as financial support throughout my entire studies. My mother, Hannalene Rossouw, for her determination, motivation and love, without which I would not have been able to complete my studies. Eben and Ilze Rossouw, my brother and sister, for keeping me sane.

Contents

List of Figures

List of Tables

List of Algorithms

List of Physical Constants

Symbol	Value	Units	Description
G	6.6728×10^{-11}	$m^3/(kg\ s^2)$	Universal gravitational constant
c	$2.997\,924\,58 \times 10^{8}$	m/s	Speed of light in a vacuum
R_e	6378136.6	m	Equatorial radius of Earth
M_e	5.972×10^{24}	kg	Approximate mass of Earth
ω_e	$7.29211505392569 \times 10^{-5}$	rad/s	Mean rotation rate of Earth
μ	$3.986\,004\,415 \times 10^{14}$	m^3/s^2	Earth gravitational parameter
μ_s	$1.327\,124\,400\,41 \times 10^{20}$	m^3/s^2	Sun gravitational parameter
μ_m	$4.902\,800\,15 \times 10^{12}$	m^3/s^2	Moon gravitational parameter
J_2	1.0826158×10^{-3}	-	Earth oblateness term
J_3	2.53881×10^{-6}	-	Earth oblateness term
J_4	1.6559×10^{-6}	-	Earth oblateness term
AU	$1.495\,978\,707 \times 10^{11}$	m	Astronomical Unit

See Appendix D.1 for the SMAD exponential atmospheric density model constants, and Appendix D.2 for the JGM-3 Earth gravity model constants.

List of Symbols

Symbol	Units	Description
x, y, z	m	Orthogonal coordinates as defined in the ECI frame (unless stated otherwise)
$\mathbf{R}_x, \mathbf{R}_y, \mathbf{R}_z$	-	Principal axis rotation matrices
R, r	m	Geocentric radius of the satellite (distance between Earth and the satellite's centres)
t_s	s	Sample time
t_{act}	minutes	Measurement period (time GPS module is active, excluding TTFF)
t_{int}	minutes	Time interval between GPS module activations
η_P	-	Relative (to permanently active GPS) power consumption factor
η_{comp}	-	Relative (to standard SGP4) computational load factor
$\mathbf{r}$	m	Satellite position vector (ECI frame)
$\mathbf{v}$ or $\dot{\mathbf{r}}$	m/s	Satellite velocity vector (ECI frame)
$\mathbf{a}$ or $\ddot{\mathbf{r}}$	m/s^2	Satellite acceleration vector (ECI frame)
$\mathbf{x}$	m and m/s	Satellite state vector (position and velocity vectors)
e	-	Eccentricity
a	m	Semi-major axis of orbit
n	rad/s	Mean Motion of a satellite
i	rad or deg	Inclination
Ω	rad or deg	Right Ascension of the Ascending Node (RAAN)
ω	rad or deg	Argument of Perigee
v	rad or deg	True Anomaly
E	rad or deg	Eccentric Anomaly
M	rad or deg	Mean Anomaly
V	-	Gravity Potential
ϕ	rad or deg	Geocentric latitude
λ	rad or deg	Geocentric longitude
B*	Earth radii^{-1}	Aerodynamic drag term in TLE
C_d	-	Drag coefficient
A	m^2	Area
ρ	kg/m^3	Atmospheric density
bc	kg/m^2	Ballistic coefficient
α_{drag}	m	Drag factor introduced for convenience ($-\frac{1}{2}\frac{\rho}{bc}$)
$F_{10.7}$	W/(m^2 Hz) or sfu	Solar radio flux index
α_t	-	In-track correction aggressiveness factor (in aSGP4)

$\mathbf{Q}$	-	System noise covariance matrix
q	$\mathrm{m}^2/\mathrm{s}^4$	Expected force model error variance
$\mathbf{R}$	-	Measurement noise covariance matrix
r_r	m^2	Expected position measurement noise variance
r_v	$\mathrm{m}^2/\mathrm{s}^2$	Expected velocity measurement noise variance

List of Abbreviations

ADCS	Attitude Determination and Control System
AP	Argument of Perigee
aSGP4	Adaptive SGP4
C	Programming Language
DCM	Direction Cosine Matrix
DoY	Day of Year
DOP	Dilution of Precision
ECEF	Earth Centred Earth Fixed Coordinate System
ECI	Earth Centred Inertial Coordinate System
EKF	Extended Kalman Filter
GAST	Greenwich Apparent Sidereal Time
GLONASS	GLObal NAvigation Satellite System (Russian)
GMST	Greenwich Mean Sidereal Time
GNSS	Global Navigation Satellite Systems
GPS	Global Positioning System
IC	Integrated Circuit
IERS	International Earth Rotation and Reference Systems Service
IIR	Infinite Impulse Response
J2000	Julian Date (year 2000 epoch)
JD	Julian Date
JGM	Joint Gravity Model
JPL	Jet Propulsion Laboratory (at NASA)
LEO	Low Earth Orbiting
MA	Mean Anomaly
MM	Mean Motion
MSIS	Mass-Spectrometer-Incoherent-Scatter
NASA	National Aeronautics and Space Administration (USA)
NAVSTAR	Navigation Satellite Timing and Ranging System
NORAD	North American Aerospace Defence Command
NSSCC	National Space Surveillance Control Center
NTW	Satellite Orbit Reference Frame (Normal, In-track and Cross-track)
OBC	On-board Computer
OD	Orbit Determination
PC	Personal Computer
PCB	Printed Circuit Board
PDOP	Position Dilution of Precision
PPT	Positions and Partials as functions of Time

PVT	Position, Velocity and Time
RAAN	Right Ascension of the Ascending Node
RK4	4rth Order Runge Kutta Method
RMS	Root Mean Square
SGP4	Simplified General Perturbations Technique (version 4)
SGPS	Space GPS receiver module
SMAD	Space Mission Analysis and Design (Textbook)
SOW	Second of Week
SRP	Solar Radiation Pressure
TLE	Two-Line Element
TTFF	Time to First Fix
UERE	User-equivalent range error
USNO	United States Naval Observatory
UT	Universal Time
UTC	Coordinated Universal Time
WN	Week Number

Chapter 1

Introduction

1.1 Purpose of this Study

1.1.1 Orbit Determination

Satellites play a vital role in today's world. Valuable human achievements like Earth observation, precise terrestrial navigation and weather prediction are realisable by satellite technology. However, none of these achievements would have been possible without orbit determination (OD) methods. Not only is this vital for ground station operation, but real time on-board knowledge of position is crucial to any space mission [1].

Over the last few decades our OD methodologies have evolved and been refined due to improvements in modelling techniques, tracking technology and computers. OD is the method of estimating the state (position and velocity, or other describing parameters) of an orbiting object [2]. This is done by developing a set of equations describing a variety of forces and using measurements to determine initial and refine later states. The complex nature of these equations leads to highly non-linear equations. This, combined with our lack of fully grasping the physics behind perturbing forces and noisy measurements, limit the accuracy of predicting a satellite's state. Today's most popular OD techniques are based on revolutionary discoveries from the 1960s to the late 1980s.

In the astrodynamics world, there are three playing fields of propagation: analytical methods (also referred to as general perturbation techniques), numerical integration procedures (also referred to as special perturbation techniques), and a middle ground called semi-analytical methods [2]. General perturbation methods superimpose the approximated effects of the most prominent perturbations on the Keplerian equations of motion. This leads to elegant and efficient propagation methods capable of predicting satellite states at any arbitrary time (within a few days) in the future or past, at the cost of lacking the small variations caused by more complex perturbations. Special perturbation methods numerically integrate orbit dynamics equations, which are derived from a more comprehensive force model. Due to the inclusion of the more complex perturbations, these methods achieve a substantial improvement in accuracy, at the price of having to propagate stepwise forward in time and a high computational load [3]. Semi-analytical methods solve the 2-body problem analytically and then numerically integrate the perturbations. The computer technology of the first few space age decades deemed analytical methods the only feasible option. The currently most popular propagator, SGP4 [1], is an analytical method. Today's computer technology has evolved to the stage where numerical methods are easily implemented on larger satellites (see Section 5.1 for a few case studies).

The earliest satellite orbit observations were performed with cameras. Since then the typical OD meth-

odologies involved the use of radar or laser ranging devices from Earth stations. These were used to derive orbit describing parameters, which were uploaded to the satellite for on-board propagation. Global navigation satellite systems (GNSS) revolutionised OD industry. USA's Global Positioning System (GPS), discussed in Appendix B, is the first and most famous of these systems. The Russian GLONASS is the only other GNSS operational today, and the European Galileo and Chinese BeiDou are two more GNSSs under construction. Satellites equipped with GPS receivers are able to accurately determine their orbit state without the aid of any ground segment. The precision of this technique is superior to propagation systems, and is thus a natural choice for missions requiring more accurate navigation solutions [1]. But implementing a dynamical orbit determination algorithm in conjunction with a GPS receiver can enhance the accuracy by even another 1-2 orders [3]. Only in the last decade or so has the technology of electronics allowed GPS receiver units to be manufactured compact enough to fit on extremely small satellites.

1.1.2 Satellite Size

The previous millennium marked a period where substantial funding was required to develop and launch a satellite. The main cause for this was the cost of launching an object with the mass and size required for a functional satellite. It was the bulky electronics of the day that prevented smaller satellite designs. Advances in technology over the last few decades has made it possible to downscale transistor size, and subsequently the size of ICs, leading to much smaller satellite designs.

In 1999, the CubeSat Project began with the purpose of streamlining accessibility to space by reducing the cost and development time of LEO nanosatellites [4]. Starting off as a collaboration between California Polytechnic State University and Stanford University's Space Systems Development Laboratory, there are currently over 100 private companies, universities and high schools involved in CubeSat projects. The idea behind the project is to define certain specifications that would streamline the design, construction and launch processes of nanosatellite missions. The principle constraints defining a 1U CubeSat are the $10 \times 10 \times 10$ cm dimensions and 1.33 kg mass limits. The CubeSat industry has evolved to include larger 1.5U, 2U, 3U and even bigger definitions. These have become the standards for most nanosatellite missions.

Satellites that are this small have exceptionally restrictive power budgets. The limited dimensions of the side panels restrict the amount area available for solar cells. Typical 10×10 cm CubeSat solar panels can yield 1.8-2.4 W when exposed to sunlight at a 90° incidence angle in LEO [5; 6]. Satellites with two or three adjacent solar panels can orientate itself so that all three point towards the Sun (albeit not at a 90° incidence angle) to increase power generation. Deployable solar panels can be used, but mechanisms are usually avoided for small satellites. Comprehensive power budgeting is not in the scope of this thesis, but it should be clear that the typical 1 W power usage of a GPS receiver (see Appendix B.9) is infeasible for such small satellites. An example is the CanX-2 3U CubeSat, which, due to power constraints, could only operate the on-board GPS receiver for a maximum of 45 minutes [7].

1.1.3 Proposed Solution

The purpose of this study is to solve the problem of obtaining precise knowledge of position on-board a nanosatellite without having the burden on the power budget of a permanently active GPS receiver. It is proposed to provide the nanosatellite an on-board GPS receiver module, but to limit the time it is powered. Orbit state estimation will be done when the GPS receiver is active, after which a propagation algorithm will be used to obtain satellite navigation solutions. As it is the GPS receiver module, and not the propagation algorithm executed by the OBC, that causes the power consumption, cutting its active time will significantly reduce the average power consumption of the on-board OD solution. Figure 1.1

visualises the proposed system: t_{act} indicates the part of the orbit where the GPS receiver is active, and t_{int} indicates the interval between GPS receiver activations.

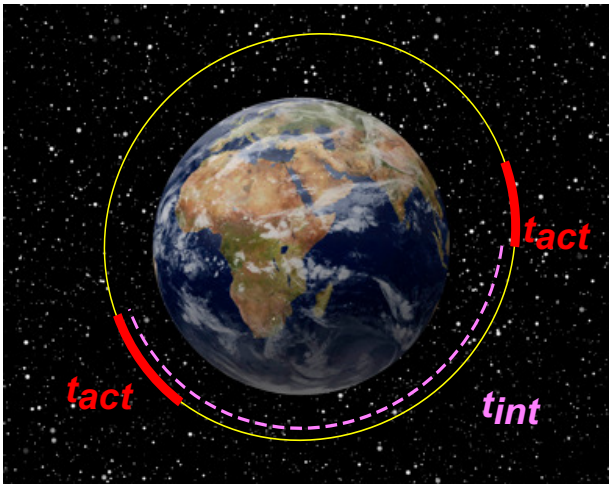


Figure 1.1 – Illustration of proposed system (image of Earth obtained from [8]).

Both an analytical and a numerical approach will be investigated. SGP4 will be used as the propagating basis for the analytical approach. The SGP4 method is extended to use GPS measurements to derive variable parameters that indicate how it has drifted from its epoch input. This approach is named the aSGP4 system. On the other end, an EKF is implemented for state estimation. A Runge-Kutta numerical integration technique is used for propagation.

1.2 Scope of this Study

1.2.1 Implementation Requirements

Before the proposed system can be developed, the physics behind LEO astrodynamics must first be extensively researched. After this, the orbit determination algorithms will be mathematically derived and implemented in the C programming language, as it is a widely used platform for developing software intended to run on microcontrollers. C is also the basis for most of the current satellite projects at the ESL, and thus portability to the in-house OBC, CubeComputer, will be straightforward.

The main deliverable of this thesis will be orbit determination code written in C. No hardware is thus required for this project, as the end product is a software module. To facilitate the implementation on real satellites, the software and interface will be kept as generic as possible.

Testing of the system will be done through simulations using position and velocity measurements that was logged by GPS receivers on board larger satellites. The simulation setup (obtaining GPS data, feeding it to the compiled solution, and plotting results) will be done in Matlab.

1.2.2 GPS Receiver and Data

The development and implementation of a GPS receiver module is not part of this thesis. An in-house GPS receiver module has already been developed for the ESL by B.J. Nortier [9] for the SumbandilaSat microsatellite. This thesis, though, is intended for nanosatellites, and thus it is assumed that a commercial-off-the-shelf (COTS) GPS receiver module will be used. These have improved drastically over the last few

years, and have become so compact that they can easily fit into a CubeSat. See Appendix B for an in-depth discussion on GPS theory, software and hardware.

The systems developed in this thesis are tested by means of simulation only. Three satellite GPS datasets, described in Table 1.1, were provided by the study leader. These datasets are recordings of the respective satellites' navigation solutions (ECEF position and velocity vectors, timestamps, as well as other parameters). ADCS information is not included in these datasets. Thus no knowledge of the orientation of the satellite, and thus its panels too, is available. This information is required to thoroughly model some orbit perturbations, like aerodynamic drag, and thus we will need to implement some simplifications to these models.

Table 1.1 – GPS datasets provided for this thesis.

Satellite	SumbandilaSat	NigeriaSat
Date	01/02/2010 and 07/02/2010	03/10/2003
Duration	260 and 95 min	230 min
t_s	30 s	10 s
GPS receiver	In-house	SSTL SGR-10
Altitude (Perigee × Apogee)	492 × 504 km	675 × 694 km
Inclination	98°	98.21°
Dimensions	0.8 × 0.65 × 0.45 m	N/A
Mass	82 kg	90.1 kg

1.2.3 SGP4 Module

Whilst the numerical integration setup was developed from scratch, the study leader provided the C source code of a functioning SGP4 software module. This module was augmented to be able to perform state correction techniques from GPS measurement inputs, as discussed in this thesis.

SGP4 requires a TLE (discussed later in Section 3.3) dataset to initialise. TLEs for this thesis were requested and obtained from Dr. T.S. Kelso through the CelesTrak website [10].

1.3 Performance Criteria

Three benchmarks are chosen to evaluate our system: accuracy, power consumption and computational load.

1.3.1 Accuracy

As with all engineering fields, an accuracy specification is a very difficult term to define and guarantee. Many specifications refer to *typical* or *expected* accuracies. To make things worse, the terms accuracy and precision are often used interchangeably. Where accuracy refers to the difference between true and measured values, precision refers to the repeatability of results under the same conditions. Figure 1.2 illustrates these concepts. Throughout this thesis, these two terms will be used interchangeably to indicate accuracy.

Since this thesis is based on data recorded from a satellite flown GPS receiver module, tests repeated under the same conditions will always return the same results, unless additional random noise is superimposed onto the GPS data. Instead, the concept of precision was accounted for by repeating our simulation with different starting points in the GPS dataset. Noise was only superimposed when the data was manipulated to suite propagator requirements (like the upsampling process described in Section 5.7.2).

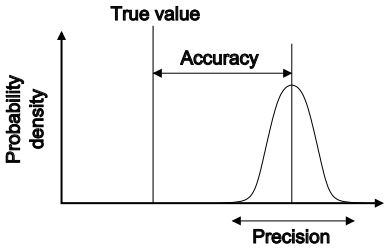


Figure 1.2 – Basic difference between accuracy and precision.

The term *error* in this thesis refers to the measured value minus the estimated value. As the precise true position of a satellite is hardly ever known, predictions can only be compared to other estimates which are known to be much more precise. It is safe to assume that the on-board GPS receiver's measurements are substantially more precise than our propagator system's predictions, and can thus be regarded as the true states when compared to our propagator's results. Studies described later in Section 5.1 used least square fittings to obtain more precise approximations of the true states, however these studies were aimed at sub-decimeter accuracy requirements.

Two types of accuracy are used to benchmark our propagation systems: *Root Mean Square* (RMS) and maximum errors. RMS is a good measurement of typical/expected accuracy of a system, and can be found with Equation 1.1.

$$\text{RMS error} = \sqrt{\sum_{k=0}^{n-1}(x_k - \hat{x}_k)} \tag{1.1}$$

where x_k is the true value, $\hat{x}_k$ is the estimated value, and n is the number of samples considered. SGP4 typically produces 3D RMS position errors on the order of 1 km upon epoch [7], whereas SumbandilaSat's GPS receiver can produce positions accurate to 10 m 3D RMS (see Appendix B.8). The maximum error is the absolute value of the single error with the greatest magnitude throughout the entire simulation.

The work of this thesis is aimed at CubeSats, which are much smaller than SumbandilaSat and NigeriaSat (which are considered microsatellites). CubeSats are most likely more susceptible to disturbances like aerodynamic drag (which is very unpredictable), and therefore our system could possibly present worse results on nanosatellites. It is assumed that the accuracies obtained with our simulations are still within the same order of what nanosatellite simulations would have produced. Our limited quantity of datasets (two of SumbandilaSat and one of NigeriaSat) also detracts the credibility of our results.

1.3.2 Power Consumption

The main goal of this project is to decrease the overall power required for navigation by reducing the time the GPS receiver needs to be active. A factor η_P is used to compare the expected power consumption of the systems developed in this thesis to that of a permanently active GPS receiver:

$$\eta_P = \frac{P_{GPS\ and\ propagator\ system}}{P_{Continuous\ GPS}} \approx \frac{t_{act} + \text{TTFF}}{t_{int}} \tag{1.2}$$

where t_{act} denotes the timespan the GPS measurements are available, t_{int} is the time interval between the GPS reactivations, and TTFF is the time to first fix. The optimal solution for this system will have the lowest t_{act} and highest t_{int} and still provide an acceptable accuracy.

1.3.3 Computational Load

As this project requires no hardware implementation, obtaining the maximum allowable computational load becomes difficult to qualify. The ways in which software components are compiled and executed vary

between different OBCs and flight software.

SGP4 is a standard propagator which has been flown on many nanosatellites. As it is an analytical propagation technique developed for computers of a few generations ago, it has a low computational load. Current nanosatellite OBCs are typically capable of running more complex orbit propagation software than SGP4. As it is aimed to implement this software system on CubeComputer, and not on its own microcontroller, it will have to share its processing time with many other software components. CubeComputer is an OBC developed by the ESL at Stellenbosch University, and contains a 32-bit ARM Cortex-M3 based microcontroller, which also services the ADCS, C&DH (command and data handling), TT&C (Telemetry, Tracking and Command), mass storage and payloads [11]. Thus we must still try to minimise the OBC's computational burden. The newly developed orbit propagator's computational load will be compared against that of the standard SGP4 propagator. All of the propagator code will be written in C and run from a Matlab script. The execution times each propagator requires for the same simulated timespan will be used to approximate how their computational loads compare. Matlab profiler will be used to determine the time that each the Simulation required to execute. For consistency, the same PC, with no other programs open and only one core dedicated to the simulation, will be used.

A computational load factor η_{comp}, defined in Equation 1.3, will be used to compare the execution times of the different techniques relative to SGP4.

$$\eta_{comp} = \frac{\text{computations of GPS+propagator system}}{\text{computations of SGP4}} \approx \frac{\bar{t}_{GPS+propagator\ system}}{\bar{t}_{SGP4}} \tag{1.3}$$

where $\bar{t}_{GPS+propagator\ system}$ is the average time the system under investigation took to finish the simulation, and $\bar{t}_{SGP4}$ is the average time the SGP4 module took. Some factors should also be considered for analysing this benchmarking method's results.

- Normal SGP4 propagation does not require regular small steps to maintain accurate results. A numerical method's accuracy, on the other hand, deteriorates rapidly when the step size is increased beyond a certain point. Unless a higher sampling rate is required, a 30 s sample time will be used for propagation.
- Propagation methods with GPS feedback differ in timespan, regularity or sample rate of GPS measurements.
- Microcontrollers have different ways of executing floating point operations and some routines (such as trigonometric functions). This might affect the η_{comp} ratios if it was measured on a microcontroller. This issue will come to light when the system is implemented on a specific OBC.
- To make the benchmarking process easier, it is assumed that the ratio between a PC and a specific microcontroller's execution times remains constant for different programs. The fact that all the propagation and estimation code is written in C and compiled before execution makes this assumption feasible. This means that the ratio between execution times of programs run on a PC will remain approximately the same when executed on microcontrollers.

1.3.4 Performance Goals

The following targets are set for this thesis:

Table 1.2 – Performance goals set for this thesis.

Specification	Target Value
η_P	< 0.15
η_{comp}	< 5
RMS position error	100 m
Max position error	500 m

1.4 Document Outline

This document is divided into chapters, with appendices providing complementary background information. Here is a brief outline of the chapters' contents:

Chapter 1 provides the purpose and scope of this thesis, and how this work is relevant to the industry. Benchmarking criteria are explained and performance goals are set.

Chapter 2 explains the physics and mathematics behind the motion of a satellite. While the principles are generic to any satellite, emphasis is placed on phenomena that have more prominent effects on LEO satellites. These phenomena include Earth's gravity and its perturbations, as well as aerodynamic drag, solar radiation pressure and third body gravitational pulls (the Sun and Moon). Forces acting upon satellites are described in an inertial Cartesian reference frame, as this is the most intuitive to understand. It is more convenient to express these forces as accelerations. Simplified and optimised routines for calculating these accelerations are provided where applicable.

Chapter 3 discusses on the industry standard SGP4 method for orbit propagation. A brief history is provided of how we arrived at our current version of this analytical method. Simulations using a provided SGP4 software module are performed and explained. The nature and cause of its errors, as well as degradation over time, are discussed.

Chapter 4 begins with the introduction of Δ parameters: a set of predetermined offsets to the SGP4 input orbital parameters to improve its accuracy. This gives insight into the effects each of these Δ parameters have. Next, a technique for automatically determining these Δ parameters, called aSGP4, is developed. It is optimised and its performance is discussed.

Chapter 5 describes another approach: an EKF based on a numerical propagator. EKF principles are first handled, after which the equations required for this specific problem are derived. The EKF is implemented and simulated. Parameter changes and some other strategies are contemplated, and finally a vastly improved EKF system is found.

Chapter 6 summarises the accomplishments of this thesis. Results are analysed and commented on. Recommendations for further research are made and issues likely to arise during implementation on real satellites are discussed.

Chapter 2

Satellite Orbit Mechanics

2.1 Introduction

Before we jump into orbit propagation techniques, we first need to understand the physics behind a satellite's motion in LEO. This is essential in order to comprehend the nature of the issues that we will encounter later on. LEO satellites in closed circular or elliptical orbits are the focus of this thesis. The starting point of this Chapter was the comprehensive *Fundamentals of Astrodynamics and Applications* [12] textbook by Vallado. *Satellite Orbits: Models, Methods and Applications* by Montenbruck [13] is also an excellent read.

We first discuss the most basic orbit description, the so-called two-body problem, and then perturbations will be handled. According to SMAD [14], the largest perturbing forces affecting LEO satellites (in order of impact) are:

1. Earth higher-order geopotential
2. Atmospheric drag
3. Solar radiation pressure
4. Sun/Moon point mass

2.2 Notation

It is recommended that Appendix A is studied before this chapter is read. It explains necessary concepts about coordinate frames and time systems that are used throughout this chapter.

We define the following vectors in the Cartesian ECI frame to indicate the position, velocity and acceleration of a satellite:

$$\begin{aligned}
\mathbf{r}(t) &= x(t)\hat{\mathbf{i}} + y(t)\hat{\mathbf{j}} + z(t)\hat{\mathbf{k}} \\
\mathbf{v}(t) &= \dot{\mathbf{r}}(t) = \frac{dx(t)}{dt}\hat{\mathbf{i}} + \frac{dy(t)}{dt}\hat{\mathbf{j}} + \frac{dz(t)}{dt}\hat{\mathbf{k}} = \dot{x}(t)\hat{\mathbf{i}} + \dot{y}(t)\hat{\mathbf{j}} + \dot{z}(t)\hat{\mathbf{k}} \\
\mathbf{a}(t) &= \ddot{\mathbf{r}}(t) = \frac{d^2x(t)}{dt^2}\hat{\mathbf{i}} + \frac{d^2y(t)}{dt^2}\hat{\mathbf{j}} + \frac{d^2z(t)}{dt^2}\hat{\mathbf{k}} = \ddot{x}(t)\hat{\mathbf{i}} + \ddot{y}(t)\hat{\mathbf{j}} + \ddot{z}(t)\hat{\mathbf{k}}
\end{aligned} \tag{2.1}$$

where the direction unit vectors $\hat{\mathbf{i}}$, $\hat{\mathbf{j}}$ and $\hat{\mathbf{k}}$ indicate the x, y and z directions of a Cartesian coordinate frame. To keep the writing concise, we usually drop the time function indication (t) as well as the Cartesian

coordinate frame direction unit vectors, and rather use the matrix notation:

$$\mathbf{r} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \qquad \dot{\mathbf{r}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \qquad \ddot{\mathbf{r}} = \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} \tag{2.2}$$

We will also often refer to the state of the satellite as the combination of its position and velocity vectors:

$$\mathbf{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{2.3}$$

The magnitude of a vector is notated and defined as:

$$|\mathbf{r}| = \sqrt{x^2 + y^2 + z^2} \tag{2.4}$$

The same principle applies to the magnitude of other vectors. This then enables us to indicate the direction of a vector by defining its unit/normalised vector (note that the vector under consideration is indicated by a subscript):

$$\mathbf{u}_r = \frac{\mathbf{r}}{|\mathbf{r}|} \tag{2.5}$$

The effect of the perturbing forces are modelled as accelerations and superimposed onto the two-body equations to form a complete force model. This gives us the acceleration a satellite undergoes as a function $\mathbf{f}$ of its current state:

$$\mathbf{f}(\mathbf{r}, \dot{\mathbf{r}}) = \ddot{\mathbf{r}}_{sat} = \ddot{\mathbf{r}}_{2-body} + \ddot{\mathbf{r}}_{Earth\ gravity\ perturbations} + \ddot{\mathbf{r}}_{aerodynamic\ drag} + \ddot{\mathbf{r}}_{3rd\ body} + \ddot{\mathbf{r}}_{SRP} \tag{2.6}$$

2.3 Two-body problem

2.3.1 Definition of Two-body Problem

Even though Newton's (1642 - 1727) work on Calculus is required to fully understand and solve the orbit problem, orbits have been described by many before his time [12]. Johannes Kepler (1571 - 1630) is the most notable historical figure in this field. Although his work was based on the motion of the planets around the Sun, the same principles can be applied to Earth orbiting satellites.

The two-body problem is the most elementary orbit description. The following assumptions are essential for this [12]:

1. The satellite's mass is negligibly small compared to the attracting body.
2. The reference frame is inertial. This removes the derivatives of the reference frame when differentiating vectors. We will use the ECI frame (a pseudoinertial frame) in this section.
3. The satellite and attracting body are spherically symmetrical and of uniform density. This allows us to treat them as point masses.
4. No other forces than the gravity between the satellite and attracting body are present.

2.3.2 Kepler's Contribution

Johannes Kepler was a strongly devout and diligent man. He was born two months prematurely, and subsequently battled poor health and suffered from poor eyesight throughout his life. In 1601, he replaced Tycho Brahe as the imperial court mathematician for Emperor Rudolph II in Prague, Czech Republic.

Brahe left Kepler with extremely precise observational data, which he later used for his discoveries. Inspired by the relatively large eccentricity of Mars' orbit, Kepler published his Astronomia Nova (New Astronomy) in 1609. This work included his first two laws. In 1619 he accomplished his lifelong goal to describe the physical motion of planets in his Harmonices Mundi Libre (Harmony of the World). However, he considered his eventual third law (probably his most notable work) and 13 other theorems, as background to Harmonices Mundi Libre. Kepler had a very modest view of his efforts, but his laws sparked the dawn of a new period of mathematics and lead directly to Newton's accomplishments. His three laws are summarised in Vallado[12]:

1. The orbit of each planet is an ellipse (or any other conic section) with the Sun at one focus.
2. The line joining the planet to the Sun sweeps out equal areas in equal times.
3. The square of the period of a planet is proportional to the cube of its mean distance to the Sun.

2.3.3 Classical Orbital Elements

The two-body orbit can be described in entire by the Classical Orbit Element Set and a time epoch. Due to the remarkable work of Kepler, the six classical orbital elements are commonly referred to as the Kepler elements. This set describes an orbit with parameters that make the geometrical interpretation intuitive. The classical orbit element set consists of 6 parameters that describe the entire satellite orbit, including where in the orbit the satellite is. The set consists of the semi-major axis (a), eccentricity (e), inclination (i), right ascension of the ascending node (Ω), argument of perigee (ω) and true anomaly (v). They are almost always associated with an epoch time. While many other satellite state representation sets exist, the Kepler set is the most intuitive to comprehend.

Figure 2.1 shows some defining parameters of an ellipse, including the semi-major axis (a). It also illustrates how an ellipse's focal points (only one focal point, F_1, shown) can be found geometrically. Perigee and apogee are defined as the nearest and farthest points in the orbit from the attracting body, respectively.

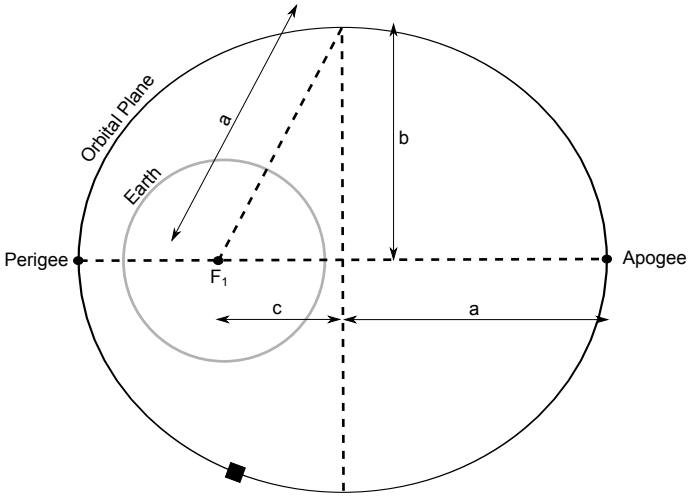


Figure 2.1 – Geometry of an elliptical orbit.

Using the dimensions of the Figure 2.1 sketch, the eccentricity of an ellipse is:

$$e = \frac{c}{a} \tag{2.7}$$

Using Figure 2.2 we can now describe the other classical elements. Inclination $i \in [0°, 180°]$ is the angle between the orbital and equatorial planes. Inclinations ranging $0° < i < 90°$ indicate a prograde orbit, i.e. the satellite's motion is in the direction of the attracting body's rotation (if Earth is the attracting body in Figure 2.2, the orbit would be prograde). $90° < i < 180°$ indicates a retrograde orbit (satellite's motion opposes the attracting body's rotation). A perfect polar orbit has $i = 90°$, and $i = 0°$ or $180°$ indicates equatorial orbits.

The ascending node is the point on the equatorial plane through which the satellite crosses from South to North. The right ascension of the ascending node $\Omega \in [0°, 360°)$ is defined as the angle between the vernal equinox and the ascending node vectors. This angle is thus measured along the Equatorial plane.

The argument of perigee $\omega \in [0°, 360°)$ and true anomaly $v \in [0°, 360°)$ are measured along the orbital plane. The angle between the perigee and ascending node vectors is the definition of ω. v is defined as the angle between the perigee and satellite current position vectors.

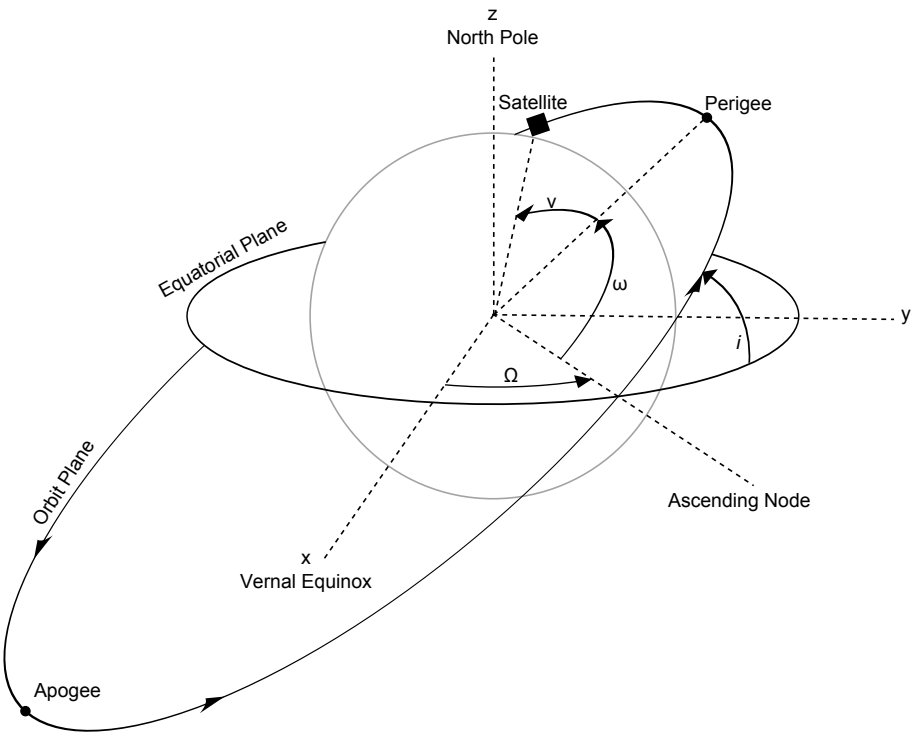


Figure 2.2 – Illustration of i, Ω, ω and v.

It is clear from the definitions above that the Kepler element set contains geometric anomalies under some special conditions. The first is when the orbit is perfectly circular ($e = 0$), ω is undefined. Secondly, if the orbit is equatorial (i.e. $i = 0$), Ω becomes undefined. Although no perfect circular or equatorial orbit ever occurs, those close to these create trouble in numerical (computer) solutions.

For an ideal 2-body problem, all the Kepler elements, except for v, are constants defining the geometry of the orbit. v is the only rapidly changing parameter. ECI Cartesian position and velocity vector parameters are all rapidly changing. Other sets like the equinoctial and canonical elements (subdivided into Poincaré and Delauny variables) are also used throughout the literature [13; 12]. These contain no singularities, and the canonical elements simplify certain calculations (as their 2-body problem state transition matrix is purely diagonal). We will not consider these systems for the work of this thesis.

2.3.4 Obtaining Orbital Elements from ECI Position and Velocity vectors

Let us define the angular momentum ($\mathbf{h}$), ascending node pointing ($\mathbf{n}$) and eccentricity ($\mathbf{e}$) vectors as:

$$\begin{aligned}\mathbf{h} &= \mathbf{r} \times \mathbf{v} \\ \mathbf{n} &= [0\ 0\ 1]^T \times \mathbf{h} = [-h_J\ h_I\ 0]^T \\ \mathbf{e} &= \frac{\mathbf{v} \times \mathbf{h}}{\mu} - \frac{\mathbf{r}}{r}\end{aligned} \tag{2.8}$$

where $r = |\mathbf{r}|$. We can simplify the eccentricity vector calculation:

$$\begin{aligned}\mathbf{e} &= \frac{\mathbf{v} \times (\mathbf{r} \times \mathbf{v})}{\mu} - \frac{\mathbf{r}}{r} \\ &= \frac{\mathbf{r}(\mathbf{v}.\mathbf{v}) - \mathbf{v}(\mathbf{v}.\mathbf{r})}{\mu} - \frac{\mathbf{r}}{r} \\ &= \left(\frac{v^2}{\mu} - \frac{1}{r}\right)\mathbf{r} - \left(\frac{\mathbf{v}.\mathbf{r}}{\mu}\right)\mathbf{v}\end{aligned} \tag{2.9}$$

where $v = |\mathbf{v}|$. Now the classical elements can be found:

$$\begin{aligned}e &= |\mathbf{e}| \\ a &= \left(\frac{2}{r} - \frac{v^2}{\mu}\right)^{-1} \\ i &= \arccos\left(\frac{h_K}{h}\right) \\ \Omega &= \arccos\left(\frac{n_I}{n}\right) && \text{if } n_J < 0 \text{ then } \Omega = 360° - \Omega \\ \omega &= \arccos\left(\frac{\mathbf{n} \cdot \mathbf{e}}{ne}\right) && \text{if } e_K < 0 \text{ then } \omega = 360° - \omega \\ v &= \arccos\left(\frac{\mathbf{e} \cdot \mathbf{r}}{er}\right) && \text{if } \mathbf{r}.\mathbf{v} < 0 \text{ then } v = 360° - v\end{aligned} \tag{2.10}$$

where $h = |\mathbf{h}|$ and $n = |\mathbf{n}|$. This method (Equations 2.8 to 2.10) is referred to as the eci2kep($\mathbf{x}$) function throughout the rest of the thesis. Note that this produces instantaneous classical elements, which are of no direct use to SGP4, which requires mean classical elements.

2.3.5 Orbital Motion

The term Kepler's Equation refers to his second law of equal areas swept out in equal time intervals [12]. Only the elliptical case is required for the purpose of this thesis. The real significance of this law is that it relates elapsed time to the angular displacement within an orbit. This is important as v does not change linearly with time unless a perfectly circular orbit is examined (which is almost never the case). v changes faster at perigee than at apogee.

To assist us, we need to introduce an angle called the eccentric anomaly, E. This is found by drawing an auxiliary circle of radius a around the orbital ellipse, and taking the angle from perigee to the intersecting point of a line extrapolated perpendicularly to the semi-major axis through the satellite position. Figure 2.3 illustrates the definition of E.

Using geometry, relating E to v and vice versa are direct processes [12]:

$$\begin{aligned}E &= \arcsin\left(\frac{\sin v\sqrt{1-e^2}}{1 + e\cos v}\right) \\ v &= \arcsin\left(\frac{\sin E\sqrt{1-e^2}}{1 - e\cos E}\right)\end{aligned} \tag{2.11}$$

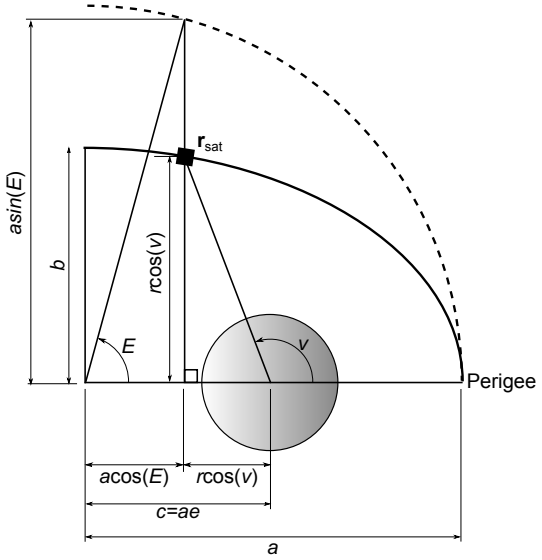


Figure 2.3 – Geometry behind Kepler's equation.

To describe the motion of a satellite in its orbit, Kepler introduced the term mean anomaly, M, which corresponds to uniform angular motion on a circle of radius a (not to be confused with E in Figure 2.3):

$$M = E - e \sin E = n(t - t_0) \tag{2.12}$$

where n, the mean motion (MM), is the mean angular rate of the orbital motion, t is the current time and t_0 is an epoch time. This is the first equation that describes the time dependence of the motion. n can be found using Kepler's third law:

$$n = \sqrt{\frac{\mu}{a^3}} \tag{2.13}$$

Obtaining M from E and e is straightforward (Equation 2.12). The reverse process, obtaining E from M and e, known generally as Kepler's Problem, is a bit more challenging. A Newton-Raphson iterative technique can be used to solve Kepler's Problem [12; 13] (not to be confused with Newton's work described in the next subsection):

Algorithm 2.1 Newton-Raphson solution to Kepler's Problem

if $-\pi < M < 0$ or $\pi < M$ **then**
  $E_0 = M - e$
else
  $E_0 = M + e$
end if
repeat
  $E_{k+1} = E_k + \dfrac{M - E_k + e \sin E_k}{1 - e \cos E_k}$
until $|E_{k+1} - E_k| <$ tolerance

2.3.6 Newton's Solution

In order to prove his ideas on mechanics, Newton needed to develop a new set of mathematical tools, namely Calculus. With this he proved that objects can be treated as point masses when solving problems of gravity.

Assuming that the Earth is a perfect sphere with a uniform density, we can use Newton's law of gravitation to determine the gravitational force f_g acting onto a satellite:

$$f_g = -\frac{Gm_e m_{sat}}{r^2} \tag{2.14}$$

where f_g is the magnitude of the gravitational force, G is a universal gravity constant, m_e and m_{sat} are the masses of Earth and the satellite, respectively, and r is the distance between the satellite and the centre of Earth. Newton's law also states that force is pointing along the line intersecting the two objects centres of mass. Then

$$\mathbf{f}_g = -\frac{Gm_e m_{sat}}{r^2}\left(\frac{\mathbf{r}}{r}\right) = -\frac{Gm_e m_{sat}}{r^3}\mathbf{r} \tag{2.15}$$

where $\mathbf{r}$ is the position vector of the satellite relative to the centre of Earth. Newton's second law of motion includes the famous relationship between an object's acceleration and the force applied to it: $F = m\ddot{r}$. We can also assume that Earth's mass remains unchanged, and thus the constants can be combined: $\mu = Gm_e$. This yields the basic two-body equation:

$$\ddot{\mathbf{r}}_{2-body} = \frac{\mathbf{f}_g}{m_{sat}} = -\frac{Gm_e}{r^3}\mathbf{r} = -\frac{\mu}{r^3}\mathbf{r} \tag{2.16}$$

Using the concepts of specific angular momentum and specific mechanical energy, one can prove that Equation 2.16 leads to all three of Kepler's laws [12].

2.4 Gravity Perturbations

2.4.1 Simple Earth Gravitation Model

The third assumption of the two-body problem cannot be made for a LEO satellite: Earth is not a perfect sphere and its mass is not uniformly spread. The dominant irregularity is the fact that Earth's equatorial radius is about 21 km longer than its polar radius. A simple method for taking this into account is including a so-called J_2 perturbation, which accounts for this oblateness problem. Vallado [12] shows the acceleration effect of this perturbation:

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}_{J_2} = \frac{3J_2\mu R_e^2}{2r^5}\begin{bmatrix}\left(\frac{5z^2}{r^2}-1\right)x\\ \left(\frac{5z^2}{r^2}-1\right)y\\ \left(\frac{5z^2}{r^2}-3\right)z\end{bmatrix} \tag{2.17}$$

Although this J_2 effect is the dominant gravity perturbation, it has many counterparts. Further modelling of irregularities is done by introducing J_3 and J_4 coefficients. The nature of their effect is illustrated in Figure 2.4. Including these are referred to as zonal expansion of a gravity model. Vallado[12] shows their resulting accelerations effects as in Equations 2.18 and 2.19.

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}_{J_3} = \frac{5J_3\mu R_e^3}{2r^7}\begin{bmatrix}\left(-3z+\frac{7z^3}{r^2}\right)x\\ \left(-3z+\frac{7z^3}{r^2}\right)y\\ \left(6z^2-\frac{7z^4}{r^2}-\frac{3r^2}{5}\right)\end{bmatrix} \tag{2.18}$$

$$\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}_{J_4} = \frac{15J_4\mu R_e^4}{8r^7}\begin{bmatrix}\left(1-\frac{14z^2}{r^2}+\frac{21z^4}{r^4}\right)x\\ \left(1-\frac{14z^2}{r^2}+\frac{21z^4}{r^4}\right)y\\ \left(5-\frac{70z^2}{3r^2}+\frac{21z^4}{r^4}\right)z\end{bmatrix} \tag{2.19}$$

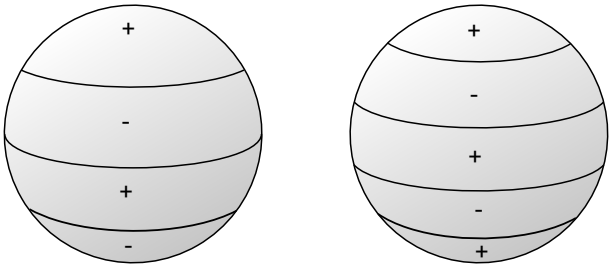


Figure 2.4 – J_3 (left) and J_4 (right) spherical harmonic expansions.

Note similarities between the x and y accelerations, which can be utilised for faster computations:

$$\ddot{y}_{J_2} = \frac{y}{x}\ddot{x}_{J_2} \qquad \ddot{y}_{J_3} = \frac{y}{x}\ddot{x}_{J_3} \qquad \ddot{y}_{J_4} = \frac{y}{x}\ddot{x}_{J_4} \tag{2.20}$$

2.4.2 Full Earth Gravity Model

A more complex model of Earth's gravity field is required for accurate orbit determination.

Several gravity models have been developed since the first satellites have been launched. These are usually described by spherical harmonic expansions, Legendre associated polynomials and a table of constants that weigh each harmonic expansion. Apart from zonal harmonics, they also include sectorial and tesseral harmonics (illustrated in Figure 2.5). These models are fixed to the topography of Earth, and therefore need to be used in the ECEF frame.

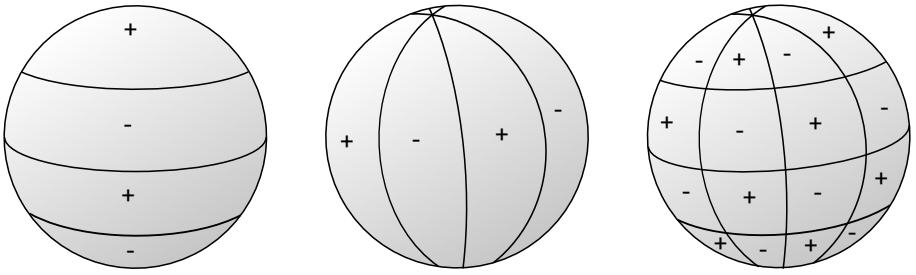


Figure 2.5 – Illustration of zonal (left), sectorial (centre) and tesseral (right) harmonics.

A popular model is the JGM-3 model, which was released in 1994 as a joint effort by NASA/GSFC (Goddard Space Flight Centre) and the University of Texas at Austin, USA [15]. As this model is widely available, it was used for this project. The full JGM-3 model (official documentation provided by [16]) has an order 70 expansion (4758 coefficients), however, this will be truncated to only the first 40×40 terms for the work of this thesis. More modern models, comprising of much more components, have been developed. The EGM2008 model has an order of 2159 (official documentation provided by [17]). The truncation margin will be discussed later.

2.4.2.1 Mathematical Description of Gravity Models

The definition of a gravitational potential, V, and the acceleration a satellites will experience under its influence, is fundamental to any gravity model:

$$\begin{aligned} V &= G \int \frac{\rho(\mathbf{s})}{|\mathbf{r}-\mathbf{s}|} d^3\mathbf{s} \\ \ddot{\mathbf{r}} &= \nabla V \end{aligned} \tag{2.21}$$

where $\rho(\mathbf{s})$ is the density at some point $\mathbf{s}$ inside Earth, and $|\mathbf{r}-\mathbf{s}|$ indicates the distance from the satellite to this point. For this subsection, position coordinates are in the ECEF frame.

In 1969, Cunningham [18] published a paper on the fundamentals of spherical harmonic expansion mathematics required for the calculation of orbital motion. The derivation is done through expanding the inverse distance using Legendre polynomials [13]:

$$\frac{1}{|\mathbf{r}-\mathbf{s}|} = \frac{1}{r}\sum_{n=0}^{\infty}\left(\frac{s}{r}\right)^n P_n(\cos\gamma) \tag{2.22}$$

where

$$\begin{aligned}\cos\gamma &= \frac{\mathbf{r}\cdot\mathbf{s}}{rs}\\ P_n(u) &= \frac{1}{2^n n!}\frac{d^n}{du^n}\left(u^2-1\right)^n\end{aligned} \tag{2.23}$$

Let's introduce longitude λ and latitude ϕ for the satellite:

$$\begin{aligned}\lambda &= \arctan\left(\frac{y}{x}\right)\\ \phi &= \arctan\left(\frac{z}{\sqrt{x^2+y^2}}\right)\end{aligned} \tag{2.24}$$

as well as λ' and ϕ' for $\mathbf{s}$. We can use an additional theorem of the Legendre polynomials [13]:

$$P_n(\cos\gamma) = \sum_{m=0}^{n}(2-\delta_{0m})\frac{(n-m)!}{(n+m)!}P_{nm}(\sin\phi')\cos(m(\lambda-\lambda')) \tag{2.25}$$

where δ_{0m} is the Kronecker delta function (in this case it is 1 if $m=0$, and 0 elsewhere), and P_{nm} is the associated Legendre polynomial of degree n and order m, defined as [13]:

$$P_{nm}(u) = (1-u^2)^{m/2}\frac{d^m}{du^m}P_n(u) \tag{2.26}$$

Subsequently, Cunningham [18] obtained the following form:

$$V = \sum_{n=2}^{\infty}\sum_{m=0}^{n}\frac{R_e^n}{r^{n+1}}P_{nm}(\sin\phi)(C_{n,m}\cos(m\lambda)+S_{n,m}\sin(m\lambda)) \tag{2.27}$$

where $C_{n,m}$ and $S_{n,m}$ are physical constants that define a specific gravity model, describing mass distribution of Earth. These constants are thus defined as (notice the independence of satellite position):

$$\begin{aligned}C_{nm} &= \frac{2-\delta_{0m}}{M_e}\frac{(n-m)!}{(n+m)!}\int\frac{s^n}{R_e^n}P_{nm}(\sin\phi')\cos(m\lambda')\rho(\mathbf{s})d^3\mathbf{s}\\ S_{nm} &= \frac{2-\delta_{0m}}{M_e}\frac{(n-m)!}{(n+m)!}\int\frac{s^n}{R_e^n}P_{nm}(\sin\phi')\sin(m\lambda')\rho(\mathbf{s})d^3\mathbf{s}\end{aligned} \tag{2.28}$$

Appendix D.2 contains a list of the JGM-3 C_{nm} and S_{nm} up to order 20.

2.4.2.2 Recursive Methods

Define

$$x_0 = x\frac{R_e}{r^2} \qquad y_0 = y\frac{R_e}{r^2} \qquad z_0 = z\frac{R_e}{r^2} \tag{2.29}$$

Let's introduce $\bar{V}_{n,m}$, a complex variable:

$$\bar{V}_{n,m} = \frac{R_e^n P_{nm}(\sin\phi)(x+jy)^m}{r^{n+m+1}\cos^m\phi} \tag{2.30}$$

then

$$V = Real \sum_{n=2}^{\infty} \sum_{m=0}^{n} (C_{n,m} - jS_{n,m}) \bar{V}_{n,m} \tag{2.31}$$

A theoretically complete model would have $n = 1, 2, 3...\infty$ and $m = 1, 2, 3...n$, but this impossible. In reality, we truncate our model: $n = 2, 3, 4...$model order, and $m = 0, 1, 2...$model degree, with order $\geq$ degree (usually, order = degree, and this convention will be adhered to). By initialising the following two terms

$$\begin{aligned} \bar{V}_{0,0} &= \frac{R_e}{r} \\ \bar{V}_{1,0} &= z_0 \bar{V}_{0,0} \end{aligned} \tag{2.32}$$

the following recursive relations (formally derived in [18; 13], but adapted here) can be used to determine the rest of $\bar{V}_{n,m}$:

$$\begin{aligned} \bar{V}_{m,m} &= (2m-1) \frac{x_0 + jy_0}{r^2} \bar{V}_{m-1,m-1} \\ \bar{V}_{m+1,m} &= (2m+1) z_0 \bar{V}_{m,m} \\ \bar{V}_{n,m} &= \left((2n-1) z_0 \bar{V}_{n-1,m} - (n+m-1) \frac{R_e^2}{r^2} \bar{V}_{n-2,m} \right) \end{aligned} \tag{2.33}$$

The process of recursively evaluating the entire V potential is illustrated in Figure 2.6. Red arrows (↘) indicate the first line of Equation 2.33, blue arrows (↓) the second line, and green (↓) the final line.

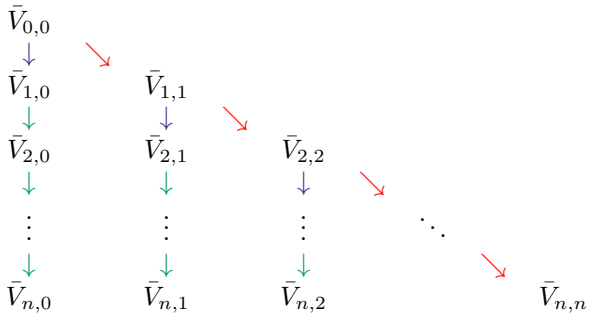


Figure 2.6 – Illustration of recursive implementation of gravity model.

Up to this point we have only discussed the gravity potential V. The acceleration experienced by a satellite caused by a gravity model requires the differentiation of its potential function. The following formulas are derived in [18]:

$$\begin{aligned} \ddot{x}_{n,m} &= \frac{\partial \bar{V}_{n,m}}{\partial x} = \begin{cases} -\frac{1}{2}\bar{V}_{n+1,m+1} + \frac{(n-m+1)(n-m+2))}{2} \bar{V}_{n+1,m-1}, & m > 0 \\ -\frac{1}{2}\bar{V}_{n+1,1} - \frac{1}{2}\bar{V}^*_{n+1,1}, & m = 0 \end{cases} \\ \ddot{y}_{n,m} &= \frac{\partial \bar{V}_{n,m}}{\partial y} = \begin{cases} j\frac{1}{2}\bar{V}_{n+1,m+1} + j\frac{(n-m+1)(n-m+2))}{2} \bar{V}_{n+1,m-1}, & m > 0 \\ j\frac{1}{2}\bar{V}_{n+1,1} - j\frac{1}{2}\bar{V}^*_{n+1,1}, & m = 0 \end{cases} \\ \ddot{z}_{n,m} &= \frac{\partial \bar{V}_{n,m}}{\partial z} = -j\frac{n-m+1}{2} \bar{V}_{n+1,m} \end{aligned} \tag{2.34}$$

2.4.2.3 Implementation

Under a project named GPSTK [19], the Applied Research Laboratories at the University of Texas at Austin, USA, has released partial C++ source code for implementing a gravity model. $\bar{V}_{n,m}$ is split into its real and imaginary parts: $\bar{V}_{n,m} = V_{n,m} + jW_{n,m}$, and a faster formula for $\bar{V}_{n,0}$ is also used. Montenbruck [13] explains the same method. The process is summarised in Algorithm 2.2.

Algorithm 2.2 Gravity model geopotential

$V_{0,0} = \frac{R_e}{r} \qquad W_{0,0} = 0 \qquad V_{1,0} = z_0 V_{0,0} \qquad W_{1,0} = 0 \qquad \rho = \frac{R_e^2}{r^2}$
for n=2 to order+1 **do**
  $V_{n,0} = \frac{1}{n}\left((2n-1)z_0 V_{n-1,0} - (n-1)\rho V_{n-2,0}\right)$
  $W_{n,0} = 0$
end for
for m=1 to order+1 **do**
  $V_{m,m} = (2m-1)\left(x_0 V_{m-1,m-1} - y_0 W_{m-1,m-1}\right)$
  $W_{m,m} = (2m-1)\left(x_0 W_{m-1,m-1} + y_0 V_{m-1,m-1}\right)$
  $V_{m+1,m} = (2m+1) z_0 V_{m,m}$
  $W_{m+1,m} = (2m+1) z_0 W_{m,m}$
  for n=m+2 to order+1 **do**
    $V_{n,m} = \frac{1}{(n-m)}((2n-1)z_0 V_{n-1,m} - (n+m-1)\rho V_{n-2,m})$
    $W_{n,m} = \frac{1}{(n-m)}((2n-1)z_0 W_{n-1,m} - (n+m-1)\rho W_{n-2,m})$
  end for
end for

Using a truncated version of Equation 2.31, we can obtain the acceleration from the complex parts of the gravity potential model:

Algorithm 2.3 Gravity model acceleration

for n=2 to order **do**
  $\ddot{x}_{n,0} = C_{n,0} V_{n+1,1}$
  $\ddot{y}_{n,0} = C_{n,0} W_{n+1,1}$
  $\ddot{z}_{n,0} = (n+1) C_{n,0} V_{n+1,0}$
  for m=0 to n **do**
    $\alpha = \frac{1}{2}(n-m+1)(n-m+2)$
    $\ddot{x}_{n,m} = \frac{1}{2}\left(-C_{n,m} V_{n+1,m+1} - S_{n,m} W_{n+1,m+1}\right) + \alpha\left(C_{n,m} V_{n+1,m-1} + S_{n,m} W_{n+1,m-1}\right)$
    $\ddot{y}_{n,m} = \frac{1}{2}\left(-C_{n,m} W_{n+1,m+1} + S_{n,m} V_{n+1,m+1}\right) + \alpha\left(-C_{n,m} W_{n+1,m-1} + S_{n,m} V_{n+1,m-1}\right)$
    $\ddot{z}_{n,m} = (n-m+1)\left(-C_{n,m} V_{n+1,m} - S_{n,m} W_{n+1,m}\right)$
  end for
end for

Summing these gives us the Earth gravity perturbations acceleration vector:

$$\ddot{\mathbf{r}}_{Earth\ geopotential} = \sum_{n=2}^{\text{order+1}} \sum_{m=0}^{\text{n}} \begin{bmatrix} \ddot{x}_{n,m} & \ddot{y}_{n,m} & \ddot{z}_{n,m} \end{bmatrix}^T \tag{2.35}$$

2.4.2.4 Normalisation of Gravity Model Parameters

The gravity model terms are sometimes given in normalised form, $\bar{C}_{n,m}$ and $\bar{S}_{n,m}$. This is because the non-normalised form spans more than ten orders of magnitude [13]. Normalisation of terms minimises the risk for over- and underflow when implementing these equations on a computer. Kuga [20] describes a

method for normalising:

$$\bar{C}_{n,m} = \begin{cases} \sqrt{(4n-2)\dfrac{(n+m)!}{(n-m)!}}C_{n,m} & m > 0 \\ \sqrt{2n-1}C_{n,0} & m = 0 \end{cases} \qquad (2.36)$$
$$\bar{S}_{n,m} = \sqrt{(4n-2)\dfrac{(n+m)!}{(n-m)!}}S_{n,m}$$

In order to be compatible with the methods above, we need the non-normalised forms, and thus we will need to reverse the process of Equation 2.36. Note that $S_{n,0} = 0$, and thus S values of order 0 do not have to be converted. As gravity model coefficients are physical constants, converting to the appropriate form can be pre-calculated and no recalculation during runtime is required.

2.4.2.5 Gravity Model Order

A simulation using an EKF (described later in Chapter 5) on the SumbandilaSat GPS data was performed. Propagation made use of the JGM-3 gravity model as described above. 519 samples, 30 seconds apart from each other, were used. After this the simulation was purely propagating until it completed a full 24 hour cycle. This ensures that the satellite underwent a full ground track, thus providing an accurate estimate of each component in the gravity model. For each degree and order of the model, the magnitude of the acceleration at that call to the gravity model function was logged using Equation 2.37.

$$\ddot{r}_{n,m,i} = \sqrt{\ddot{x}^2_{n,m,i} + \ddot{y}^2_{n,m,i} + \ddot{z}^2_{n,m,i}} \qquad (2.37)$$

The average contributions of each order's zonal, tesseral and sectoral parts are shown in Figure 2.7. The most evident observation to make here is that higher order gravity model components are less significant than lower order components. As this graph is presented on a semi-log scale, a linear trend on the graph translates to an exponential relation of order to magnitude.

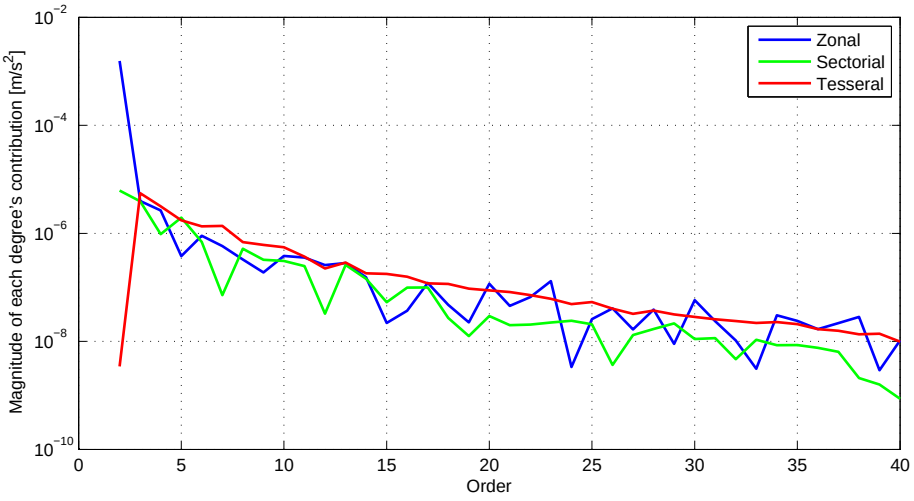


Figure 2.7 – Average acceleration that each order of the JGM-3 model contributes

One would expect the zonal harmonics to dominate the effect of the tesseral and sectorial components. Although this is true for the J_2 component (order = 2), it is not generally so for the rest of the orders. If

it was the case for the rest of the orders, one could have considered using only zonal components (or more zonal than tesseral and sectorial components) to save computation time for the same accuracy.

The nested for loop in Algorithm 2.2 indicates that computation load is proportional to the square of the gravity model's order. One must search for a balance between accuracy desired and computational effort.

2.5 Aerodynamic Drag

Even though the basic acceleration equation for aerodynamic drag is very simple, the modelling of its defining parameters is not. In fact, compared to other perturbations, aerodynamic drag dominates in uncertainty for LEO satellites. The simple formula for acceleration is [12]:

$$\begin{aligned}\ddot{\mathbf{r}}_{drag} &= -\frac{1}{2}\frac{\rho}{bc}|\mathbf{v}_{aero}|\mathbf{v}_{aero} \\ &= \alpha_{drag}|\mathbf{v}_{aero}|\mathbf{v}_{aero}\end{aligned} \tag{2.38}$$

where

ρ = atmospheric density

bc = ballistic coefficient $= \dfrac{M}{C_d A} = \dfrac{\text{mass of satellite}}{\text{drag coefficient} \times \text{cross sectional area of satellite facing } \mathbf{v}_{aero}}$

$\mathbf{v}_{aero}$ = velocity vector of the satellite relative to the atmosphere at the specific location

$\alpha_{drag} = -\dfrac{1}{2}\dfrac{\rho}{bc}$ = factor used for convenience

ρ, bc and $\mathbf{v}_{aero}$ vary throughout the orbit. A few reasons for this are [12]:

Altitude: Generally, the further you are away from Earth's surface, the less dense the atmosphere will be. There are various models to describe this decrease in density, but the most basic are that of an exponential decrease with coefficient sets for different altitude intervals (described in Subsubsection 2.5.1.1).

Diurnal variations: The local (solar) time and the season at the satellites current location influence the temperature of the atmosphere. A higher local temperature expands the air volume at the local position, pushing air upwards, and effectively increasing the density at the higher altitudes. This function is not symmetrical: the maximum appears to be around 14:00 local time, whereas minimum at around 04:00.

Solar cycle: There are two main solar activity cycles: the long 11-year cycle and the shorter 27-day cycle. The solar activity does, however, also exhibit daily fluctuations.

Satellite attitude: The orientation of the satellite relative to the air velocity vector affects the ballistic coefficient. A full drag model would include every panel of the satellite. As the satellite changes its orientation, each panel would have a varying cross-sectional area (and angle of attack) facing the velocity vector. ADCS information, however, is not available for this study, and therefore we assume that using the panel with the maximum area and a zero angle of incidence is the next best approach.

High altitude winds: At high altitudes, winds can reach extreme speeds. These do have an effect on the true air to satellite relative velocity, but it is considered insignificant in the grand scheme of things, and is subsequently ignored.

2.5.1 Atmospheric Density

Several committees have been established to create models of the atmosphere's density, for example the International Civil Aviation Organisation (ICAO), the Committee on Space Research (COSPAR), and the Committee on Extension to the Standard Atmosphere (COESA)[21]. Some of the earliest developed models include the U.S. Standard Atmosphere and the Jacchia Reference Atmosphere, which were revised and extended later on. Today, with the help of satellite orbit decay data, more complex models, like the Mass-Spectrometer-Incoherent-Scatter (MSIS) and its extensions, have been developed. Some of these require a large set of parameters.

Atmospheric density varies mostly due to altitude and temperature. Temperature highly depends on satellite position and the solar activity. It is very difficult to develop a system that can model the precise effects that these parameters have on density. From sea level to about 10 km from Earth's surface, the atmospheric temperature gradually decreases. Between 10 km and about 45 km, however, the temperature increases. Up and to about 95 km, it decreases again, and up to about 400 km it increases, whereafter it remains constant. It is clear that this type of behaviour complicates the modelling of the atmosphere. Figure 2.8 illustrates how the NRLMSISE-00 model profiles atmospheric temperature and density for typical LEO altitudes.

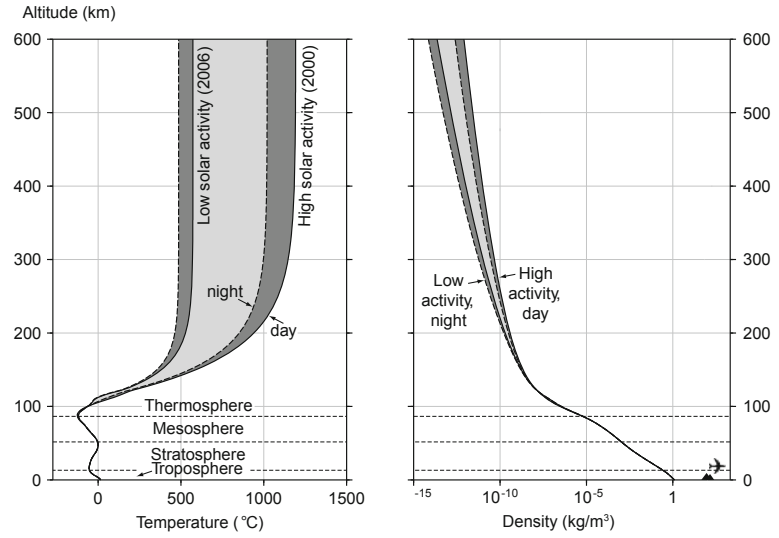


Figure 2.8 – Plot of atmospheric temperature (left) and density (right) vs altitude over Delft, The Netherlands. This is according to the later discussed NRLMSISE-00 model. Obtained from Doornbos [22].

High altitude atmospheric temperature also varies due to solar activity. This is why so many density models depend on the shorter wavelength flux measurements, an indication of solar activity [23]. The $F_{10.7}$ index is a 10.7 cm solar radio flux measurement produced daily by, amongst others, the Canadian National Research Council's Herzberg Institute of Astrophysics [24]. It is physically measured in $\mathrm{W/m^2/Hz}$, but more conveniently expressed as solar flux units ($1\ \mathrm{sfu} = 1 \times 10^{-22}\mathrm{W/m^2/Hz}$). Typical values range from 70 to 300 sfu [12]. Most density models also use an average (smoothed) value over a range of previous days. This range differs between models, but 3, 27, and 81 day averages are typical. The last two periods are due to the Sun's 27 day rotation time. Although this specific frequency of radiation represents only a small percentage of the total power received from the Sun, it has a significant effect on Earth's upper atmosphere. Figure 2.9 shows a short history of the $F_{10.7}$ index and how it is linked to atmospheric density.

Another major influence on atmospheric temperature is the diurnal effect. It is essentially the effect of the local solar time: the maximum temperature (the maximum bulge) appears to occur between 14:00

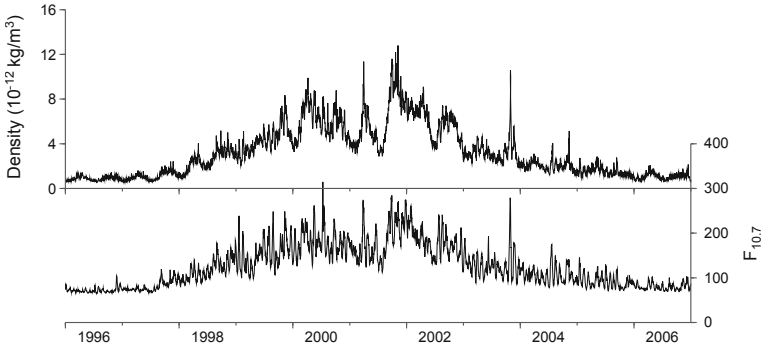


Figure 2.9 – Plot of $F_{10.7}$ index and how it is linked to atmospheric density (at 400 km altitude above Delft, The Netherlands). Obtained from Doornbos [22].

and 15:00. Some early and primitive density models used a symmetrical (about the maximum bulge) system to account for this. We know, however, that this is not the case, and many modern density models incorporate some asymmetrical function [25].

A few empirical atmospheric density model types are discussed:

1. A simple exponential decrease described by SMAD [14]
2. Jacchia
3. Harris-Priester
4. MSIS

2.5.1.1 Simple Exponential Model

The New SMAD [14] approximates the atmospheric density using an exponential model. Different sets are available for minimum, mean and maximum solar activity categories. Figure 2.10 shows how this models the atmospheric density against altitude. A simple exponential formula is used to calculate the atmospheric density:

$$\rho(h) \approx \rho_0(h) \exp\left(\frac{-(h-h_0)}{H}\right) \tag{2.39}$$

where h is the satellite altitude (in km), $\rho_0(h)$ is the nominal density (kg/m^3), h_0 is the base altitude, and H is the scale height. The values to be used for various altitude ranges are listed in Appendix D.1.

2.5.1.2 Jacchia Models

The observation of aerodynamic drag's effect on satellites' orbits paved the way for the first atmospheric density models in the 1950's [22]. The first revisions of the Jacchia Model are examples. In 1972, the Jacchia-71 model was accepted by the COSPAR International Reference Atmosphere, and is also referred to as CIRA-72. Several later revisions of the Jacchia model were published in the 1970's, but these did not introduce meaningful accuracy improvements. The US Air Force Space Command recently improved the Jacchia Model by implementing new algorithms for semi-annual and geomagnetic variations. These were released as JB2006 and JB2008.

The Jacchia-Roberts model divides the atmosphere into three bands: 90-100 km, 100-125km, and above 125 km [25]. Each band's initial conditions are determined by its predecessor's terminating values. Thus to calculate the atmospheric density at an altitude in a certain band requires the calculation of the final

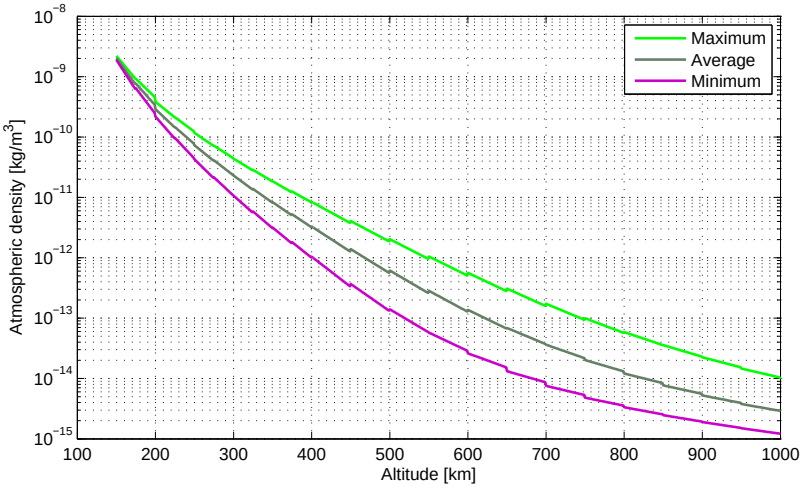


Figure 2.10 – Estimated atmospheric densities at high altitudes using SMAD exponential model.

densities of all the lower bands. The first two bands require the integration of the barometric equation and the diffusion equation, respectively. These altitudes are too low and do not fall within the scope of this thesis.

In the Jacchia range of models, the exospheric temperature T_∞ accounts for the short term density variations. T_∞ is a function of the solar flux index, diurnal effect, and geomagnetic activity. The density obtained from T_∞ is then adjusted to account for the seasonal and diurnal effects. An asymmetric temperature distribution is computed for the diurnal effect.

The overarching algorithm of CIRA-72 is described below [22]:

$$\begin{aligned} T_c &= \alpha + \beta \bar{F}_{10.7} + \gamma(F_{10.7} - \bar{F}_{10.7}) \\ T_\infty(t,\phi,\lambda) &= T_c(t) D(t,\phi,\lambda) + \Delta T_G(K_p) \\ T_x(T_\infty) &= a + bT_\infty + ce^{kT_\infty} \end{aligned} \tag{2.40}$$

T_c is the culmination point temperature, which is the minimum exospheric temperature over the whole globe when the geomagnetic index (K_p) is zero. It is obvious that this will occur at local night-time. Calculating T_c also requires the solar activity index, $F_{10.7}$, and its 81-day average, $\bar{F}_{10.7}$, using the three empirical constants α, β and γ.

The local exospheric temperature T_∞ is calculated by multiplying T_c with a diurnal variation function, $D(t,\phi,\lambda)$ and adding correction for geomagnetic activity, $\Delta T_G(a_p)$. $D(t,\phi,\lambda)$ defines the shape of the diurnal bulge, and is shown in Figure 2.11. It is a smooth function that varies between T_c and 1.3 times T_c.

The temperature profile is further constrained to a boundary condition of $T = 183$ K and $dT/dh = 0$ at 90 km altitude, with an inflection point at 125 km. T_x, the inflection point temperature is derived from T_∞ and four empirical constants a, b, c and k. The number density can now be calculated from the complete temperature profile by integrating the diffusion equation.

This method depends on the latest $F_{10.7}$ values. Requiring the regular upload of parameters compromises the automation of this method. Ideally the propagator software should be as independent of the ground segment as possible.

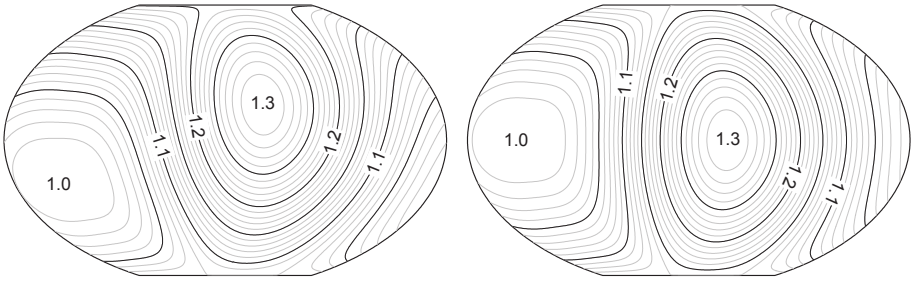


Figure 2.11 – CIRA-72 diurnal variation functions mapped to a Winkel-Tripel projection in local solar time and latitude. Left shows the function at solstice, while right shows it at equinox. Obtained from [22].

2.5.1.3 Harris-Priester Models

The Harris-Priester system was one of the earliest atmospheric density models to be developed. Its more commonly used variation, the Modified Harris-Priester Model, is based on large static tables, with ranges from 120-800 km. The NASA Goddard Space Flight Centre expanded the table to reach 100 - 1000 km [25].

The Modified Harris-Priester Model used ten separate tables for different 5 month solar flux average ranges. Each of these tables are then divided into twelve subtables which listed atmospheric densities for 2-hour intervals of local solar time. It is already clear that such an model would require an enormous amount of memory to be implemented [25].

The atmospheric density is determined by obtaining the maximum and minimum entries associated with the $\bar{F}_{10.7}$ and altitude, and then applying the following cosine function (lagging 30° from the subsolar point) for the diurnal effect [25]:

$$\rho(h) = \rho_{min}(h) + \big(\rho_{max}(h) - \rho_{min}(h)\big)\cos^n\left(\frac{\tau}{2}\right) \tag{2.41}$$

where τ is the geocentric angle between the vectors of the diurnal bulge and the satellite location. This would yield a symmetric diurnal effect. A so called Asymmetrical Modified Harris-Priester Model also exists, and takes on the following form [25]:

$$\begin{aligned}
&\rho(h) = \rho_N(h) + \big(\rho_D(h) - \rho_N(h)\big)\cos^n\left(\frac{\tau}{2}\right)\\
&\rho_N(h) = \rho_{min}(h) + \big(\rho_{max}(h) - \rho_{min}(h)\big)\sin^m(\theta)\\
&\rho_D(h) = \rho_{min}(h) + \big(\rho_{max}(h) - \rho_{min}(h)\big)\cos^m(\eta)\\
&\theta = \frac{1}{2}|\phi + \delta|\\
&\eta = \frac{1}{2}|\phi - \delta|\\
&\tau = H + \beta + \lambda\sin(H + \gamma), \qquad (-\pi \leq \tau \leq \pi)
\end{aligned} \tag{2.42}$$

where H is the local solar time, ϕ is the latitude of the subsatellite point, and δ is the solar declination. To determine δ, we need the unit vector of the Sun, as calculated in Equation 2.59 in the SRP section.

The solar declination is then:

$$\begin{aligned}
\delta &= \frac{u_{Sun,z}}{\sqrt{u_{Sun,x}^2 + u_{Sun,y}^2}} \\
&= \frac{\sin\epsilon \sin\lambda_{ecl}}{\sqrt{\cos^2\lambda_{ecl} + \cos\epsilon^2 \sin^2\lambda_{ecl}}} \\
&= \frac{\sin\epsilon \sin\lambda_{ecl}}{\sqrt{\cos^2\lambda_{ecl} + \sin^2\lambda_{ecl} - \sin^2\lambda_{ecl} + \cos\epsilon^2 \sin^2\lambda_{ecl}}} \\
&= \frac{\sin\epsilon \sin\lambda_{ecl}}{\sqrt{1 + \sin^2\lambda_{ecl}(\cos\epsilon^2 - 1)}} \\
&= \frac{\sin\epsilon \sin\lambda_{ecl}}{\sqrt{1 - \sin^2\epsilon \sin^2\lambda_{ecl}}} \\
&= \tan(\arcsin(\sin\epsilon \sin\lambda_{ecl}))
\end{aligned} \tag{2.43}$$

where $\sin\epsilon$ and $\sin\lambda_{ecl}$ is calculated as in Equation 2.59. The final step in Equation 2.43 requires more executions, and thus the previous step will be implemented. The following parameters can be chosen to fit the Harris-Priester model to the Jacchia model:

$$m = 2.2, \qquad \beta = -37°, \qquad \gamma = 43°, \qquad n = 3, \qquad \lambda = 6° \tag{2.44}$$

2.5.1.4 MSIS

Alan Hedin started developing the MSIS class of density models in the 1970's [22]. Mass spectrometer and incoherent scatter (hence the abbreviation) radar observations were the sole basis for these models. The MSIS-86 version replaced Jacchia 71 as the COSPAR International Reference Atmosphere thermosphere model, and hence is also known as CIRA-86. MSISE-90 is an extension which is identical to MSIS-86, but extends down to zero altitude. In the late 1990's, further development of MSIS was conducted by the US Naval Research Laboratory (NRL), and the NRLMSIS-00 model was published.

Along the MSIS series, the HWM (Horizontal Wind Model) series of models were also developed [22]. These model the thermospheric horizontal wind speeds, and uses the same input parameters as the MSIS models.

In the MSIS models, the exospheric temperature is calculated from solar and geomagnetic activity, just as with the Jacchia range [22]. The density, however, is not calculated by integrating altitude over the temperature profile. MSIS models actually have separate models for each of the atmospheric constituents (hydrogen, helium, oxygen, etc.). The modelled concentrations and thermospheric temperatures are calculated using a function that contains a large number of terms. Some of the terms include polynomial terms for solar and geomagnetic activity proxies and indices, spherical harmonic terms in geodetic latitude and local solar time, spherical harmonic terms in geomagnetic latitude and geomagnetic local time, periodic terms for day of year and time of day, and cross terms for solar and geomagnetic activity. Each atmospheric constituent has its own set of these coefficients. The result is a huge set of terms: for example, the NRLMSISE-00 requires about 2200 coefficients.

2.5.1.5 Atmospheric Density for this Thesis

To maintain automation and simplicity for our special perturbations approach, we will implement the Jacchia model, but use the SMAD exponential model to obtain the maximum and minimum densities.

2.5.2 Aerodynamic Velocity

We assume that the atmosphere rotates at the same rate as Earth. Vallado [12] states that this rotation might be true for near Earth atmosphere (due to friction) but that the upper parts might rotate slightly slower. From [26] we can calculate $\mathbf{v}_{aero}$ (in ECI) using an Earth fixed rotating atmosphere:

$$\mathbf{v}_{aero} = \begin{bmatrix} \dot{x}_{aero} \\ \dot{y}_{aero} \\ \dot{z}_{aero} \end{bmatrix} = \begin{bmatrix} \dot{x} + \omega_e y \\ \dot{y} - \omega_e x \\ \dot{z} \end{bmatrix} \tag{2.45}$$

2.5.3 Typical Acceleration Magnitude: SumbandilaSat

In this subsection we will determine the typical magnitude of aerodynamic acceleration. This is done only to be able to compare aerodynamic drag's significance to other perturbation's.

For the sake of simplicity, the atmospheric density and ballistic coefficients were taken as mean constants. Figure A.3 (in Appendix A) shows that SumbandilaSat's altitude varied between 486 and 512 km. Using the SMAD exponential model, maximum solar activity and lowest altitude of 486 km, we obtain an atmospheric density of:

$$\rho_{SumbandilaSat} = 4.05 \times 10^{-12} e^{\frac{-(486-450)}{63.5}} = 2.297 \times 10^{-12} \text{ kg/m}^3 \tag{2.46}$$

From Figure A.3 we see that the maximum ECEF velocity is about 7705 m/s. We also know that SumbandilaSat's dimensions are $0.8 \times 0.65 \times 0.45$ m and has a mass of 82 kg. Using the maximum area and atmospheric density, and a ballistic coefficient of 2.2, we obtain an estimated drag acceleration of:

$$\begin{aligned} |\ddot{\mathbf{r}}_{drag}|_{max} &= \left| -\frac{1}{2} \cdot \frac{\rho}{bc} |\mathbf{v}_{aero}| \mathbf{v}_{aero} \right| \\ &= \frac{1}{2} \frac{\rho C_d A}{M} |\mathbf{v}_{aero}|^2 \\ &\approx \frac{1}{2} \cdot \frac{(2.297 \times 10^{-12})(2.2)(0.8 \times 0.65)}{(82)} (7705)^2 \\ &= 9.512 \times 10^{-7} \text{m/s}^2 \end{aligned} \tag{2.47}$$

This clearly puts the maximum atmospheric drag acceleration well below that of the gravity model's high order expansions. Even though this seems negligible, it must be noted that this force can increase rapidly for satellites orbiting at lower altitudes. Another aspect to consider is the manner in which drag affects the orbit: it is a non-conservative force that makes the orbit lose energy (depletion of the orbit's semi-major axis). The higher order gravity model forces' effects are periodic and do not change the overall energy of the orbit. The non-conservative aspect of drag also has a noteworthy effect on the **F** matrix as discussed in section 5.4.4.

2.6 Third Body Perturbations

Third-body perturbations only present noticeable effects on satellites with higher altitudes, and thus we will only discuss the two major sources: the Sun and the Moon. Vallado [12] describes the acceleration due to a third body as:

$$\ddot{\mathbf{r}}_{3rd\ body} = \mu_3 \left(\frac{\mathbf{r}_{sat,3}}{r_{sat,3}^3} - \frac{\mathbf{r}_{e,3}}{r_{e,3}^3} \right) \tag{2.48}$$

where $\mathbf{r}_{sat,3}$ and $\mathbf{r}_{e,3}$ are the distance vectors from the satellite and Earth, respectively, to the 3rd body, and μ_3 refers to the gravity constant of the 3rd body. This form presents some implementation difficulties.

If the Sun is the third body, the distances to the Earth and the satellite will be very similar. These distances are also very large, and having the cube of them as denominators can produce errors during numerical calculation. The numerical instability was encountered when Equation 2.48 was implemented. An alternative approach is to use a Taylor series expansion [12]:

$$\ddot{\mathbf{r}}_{3rd\ body} \approx \frac{\mu_3}{r_{e,3}^3}\left(\mathbf{r}_{e,sat} - 3\left(\frac{\mathbf{r}_{e,sat}\cdot\mathbf{r}_{e,3}}{r_{e,3}^2}\right)\mathbf{r}_{e,3} - 15\left(\frac{\mathbf{r}_{e,sat}\cdot\mathbf{r}_{e,3}}{r_{e,3}^2}\right)^2\mathbf{r}_{e,3}\ldots\right) \tag{2.49}$$

This form is numerically stable, but it is only an approximation. When we use this method for the Moon, the distances from Earth to the Moon and the satellite is much closer than that to the Sun, and we may need to include more terms of the expansion. We will use this method instead of the previous one for its the numerical stability.

2.6.1 Sun Position

From the U.S. Naval Observatory[27] and Vallado [12] we get the following equations to obtain the approximate ECI position vector of the Sun, $\mathbf{r}_s$:

$$\begin{aligned}
T_{UT1} &= \frac{JD_{UT1} - 2451545.0}{36525}\\
\lambda_s &= 280.4606184^\circ + 36000.77005361T_{UT1}\\
M_e &= 357.5277233^\circ + 35999.05034T_{UT1}\\
\lambda_{ecl,s} &= \lambda_s + 1.914666471\sin M_e + 0.019994643\sin 2M_e\\
\epsilon &= 23.439291^\circ - 0.00130042T_{UT1}\\
r_s &= 1.00140612 - 0.016708617\cos M_e - 0.000139589\cos 2M_e \text{ AU}\\
\mathbf{r}_s &= r_s\begin{bmatrix}\cos\lambda_{ecl,s}\\ \cos\epsilon\sin\lambda_{ecl,s}\\ \sin\epsilon\sin\lambda_{ecl,s}\end{bmatrix}
\end{aligned} \tag{2.50}$$

where λ_s is the mean longitude of the Sun in degrees, M_e is the mean anomaly of the Sun in degrees, $\lambda_{ecl,s}$ is the ecliptic longitude of the Sun (adjusted for aberration), ϵ is the mean obliquity of the ecliptic, and r_s is the distance between Earth and the Sun.

2.6.2 Moon Position

An algorithm to obtain the approximate Moon coordinates is provided by Vallado [12]:

$$\begin{aligned}
\lambda_{m,ecl} &= 218.32^\circ + 481267.8813T_{UT1} + 6.29\sin(134.9 + 477198.85T_{UT1})\\
&\quad - 1.27\sin(259.2 - 413335.38T_{UT1}) + 0.66\sin(235.7 + 890534.23T_{UT1})\\
&\quad + 0.21\sin(269.9 + 954397.7T_{UT1}) - 0.19\sin(357.5 + 35999.05T_{UT1})\\
&\quad - 0.11\sin(186.6 + 966.404T_{UT1})\\
\phi_{m,ecl} &= 5.13^\circ\sin(93.3 + 483202.03T_{UT1}) + 0.28\sin(228.2 + 960400.87T_{UT1})\\
&\quad - 0.28\sin(318.3 + 6003.18T_{UT1}) - 0.17\sin(217.6 - 407332.2T_{UT1})\\
\epsilon &= 23.439291^\circ - 0.0130042T_{UT1}\\
P &= 0.9508^\circ + 0.0518\cos(134.9 + 477198.85T_{UT1}) + 0.0095\cos(259.2 - 413335.8T_{UT1})\\
&\quad + 0.0078\cos(235.7 + 890534.23T_{UT1}) + 0.0028\cos(269.9 + 954397.7T_{UT1})
\end{aligned} \tag{2.51}$$

$$
\begin{aligned}
r_m &= \frac{R_e}{\sin P} \\
\mathbf{r}_m &= r_m \begin{bmatrix} \cos\phi_{m,ecl}\cos\lambda_{m,ecl} \\ \cos\epsilon\cos\phi_{m,ecl}\sin\lambda_{m,ecl} - \sin\epsilon\sin\phi_{m,ecl} \\ \sin\epsilon\cos\phi_{m,ecl}\sin\lambda_{m,ecl} - \cos\epsilon\sin\phi_{m,ecl} \end{bmatrix}
\end{aligned} \tag{2.51}
$$

where $\lambda_{m,ecl}$, $\phi_{m,ecl}$ and P are the Moon's ecliptic latitude, longitude and parallax, respectively. Note that the distance between the Earth and the Moon, as indicated here by r_m, is measured in m.

2.6.3 Typical Acceleration Magnitude

The maximum 3rd body acceleration will occur when the satellite is precisely aligned between Earth and the 3rd body, as this is where the ratio between the satellite and Earth position vectors relative to the 3rd body is the greatest. When the Earth is aligned precisely between the satellite and the 3rd body, the minimum possible acceleration will be experienced by the satellite. Using Equation 2.49, an epoch of 1 July 2015, a 500 km altitude and the geometrical cases described above, we determined the following expected acceleration ranges due to the 3rd bodies:

$$
\begin{aligned}
\ddot{r}_{Sun} &\in [1.194 \times 10^{-6}, 1.195 \times 10^{-6}] \\
\ddot{r}_{Moon} &\in [1.373 \times 10^{-6}, 1.844 \times 10^{-6}]
\end{aligned} \tag{2.52}
$$

2.7 Solar Radiation Pressure (SRP)

As the photons from the Sun have a velocity (moving at the speed of light) and have mass, they contain momentum. This means that when they hit the satellite, an impulse is created as they bounce off into other directions. A force in the opposite direction to the Sun is applied, and thus the satellite will accelerate away from the sun. Photons have a very small mass, and subsequently this force is minute and only really important for satellites at high altitudes or with a substantial surface area to mass ratio.

This section only deals with the effect of direct sunlight. A comprehensive radiation pressure model will also account for the Earth albedo and infrared radiation. Albedo is the solar radiation that is reflected from the Earth's surface, and is about 30 % of the incoming solar radiation. Earth radiates infrared photons (equivalent to about 17 % of the direct solar radiation) even in the night time. As these effects are difficult to model, and are likely to have insignificant effects, they will not be discussed.

We will make two audacious assumptions: (1) the Earth shadow function is mostly cylindrical and (2) the solar pressure magnitude can be considered constant. Satellites deploying solar sails, or in higher orbits than LEO, require a much more complex SRP model. As uncertainties in aerodynamic drag will govern all of the disturbances, a very precise SRP model is not required.

2.7.1 Simple SRP Model

Proper SRP modelling involves the analysis of which frequencies and energy levels reach the Earth. This varies with time, and thus a solar radiation constant (also called solar flux) is often used [12]:

$$
SF = 1353 \text{ W/m}^2 \tag{2.53}
$$

A time-varying formula approximating the solar flux is also given as [12]:

$$
SF = \frac{1358}{1.004 + 0.0334\cos\left(\frac{360}{365.25}D_{aphelion}\right)} \text{ W/m}^2 \tag{2.54}
$$

where $D_{aphelion}$ is the days since Earth was last at aphelion (varies with time, but usually around July 4). For simplicity (as this force will be minor) we will only use the constant value.

We must now determine the amount of momentum these photons have. Using Einstein's $E = mc^2$ energy-mass relating equation, we can find the momentum mc as [12]:

$$mc = \frac{E}{c} \tag{2.55}$$

This leads to the solar radiation pressure, or change in momentum, p_{SR}:

$$p_{SR} = \frac{SF}{c} = \frac{1353 \text{ W/m}^2}{3 \times 10^8 \text{ m/s}} = 4.51 \times 10^{-6} \frac{\text{W s}}{\text{m}^3} = 4.51 \times 10^{-6} \frac{\text{N}}{\text{m}^2} \tag{2.56}$$

and this is how SMAD [14] obtained its simple constant acceleration of

$$a_{SRP} = 4.5 \times 10^{-6} \frac{A}{m} \text{ m/s}^2 \tag{2.57}$$

where A is the cross-sectional area of the satellite normal to the Sun and m is the satellite's mass. SMAD mentions that this only applies to an absorbing surface. Introducing the effect of reflectivity and a shadowing function, we create a simple SRP acceleration model:

$$\ddot{\mathbf{r}}_{SRP} = -4.51 \times 10^{-6} \frac{A(1+r)}{m} s_{SRP} \mathbf{u}_s \tag{2.58}$$

where r is the reflection factor, typically ± 0.4 (-1 = translucent, 0 = total absorption, 1 = total reflection), s_{SRP} is a simple Earth shadow function (zero when the satellite is in Earth's shadow and 1 elsewhere), and $\mathbf{u}_s$ is the unit vector pointing to the Sun in the ECI frame. $\mathbf{u}_s$ can be found using the same parameters as Equation 2.50:

$$\mathbf{u}_s = \begin{bmatrix} \cos \lambda_{ecl,s} \\ \cos \epsilon \sin \lambda_{ecl,s} \\ \sin \epsilon \sin \lambda_{ecl,s} \end{bmatrix} \tag{2.59}$$

2.7.2 Simple Earth Shadow Function

A simple s_{SRP} function used is described by Equation 2.60 (see figure 2.12 for visual aid):

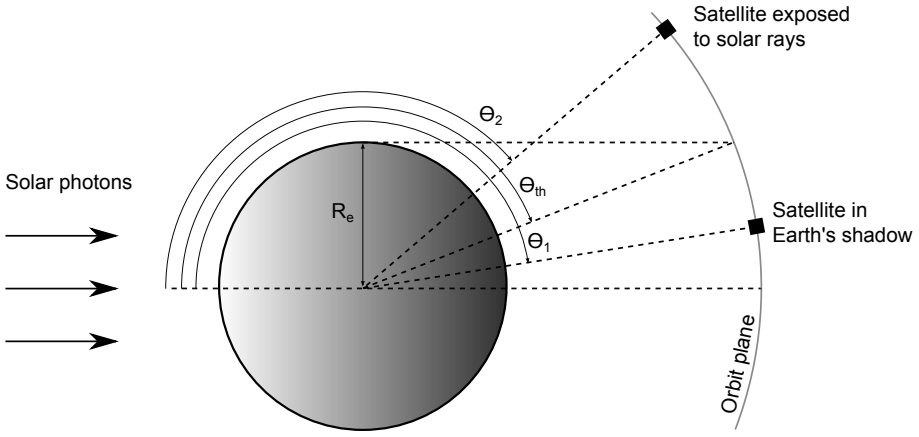


Figure 2.12 – Illustration of simple Earth shadow function.

$$\begin{aligned}\theta &= \text{acos}\left(\frac{\mathbf{u}_s \cdot \mathbf{r}_{sat}}{|\mathbf{r}_{sat}|}\right)\\ \theta_{th} &= \pi - \text{acos}\left(\frac{R_e}{|\mathbf{r}_{sat}|}\right)\\ s_{SRP} &= \begin{cases}1 & \theta \leq \theta_{th}\\ 0 & \text{else}\end{cases}\end{aligned} \tag{2.60}$$

where θ_{th} is a threshold angle. More complex shadowing functions are conic shaped [12], and differentiate between Umbra and Penumbra regions, as illustrated in Figure 2.13. As SRP has a relatively minor effect on our small satellite, the simple cylindrical approach will suffice.

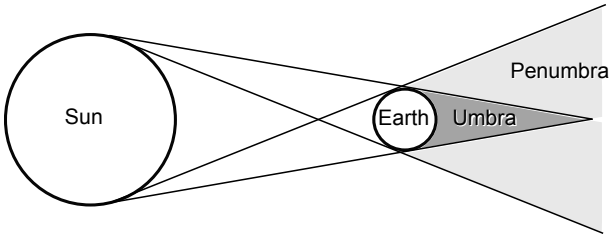


Figure 2.13 – Illustration of conic shaped Earth shadow function.

The SRP force direction is mostly normal to the Sun exposed surface, but a minor tangential component also exists. Without attitude knowledge, modelling this force in terms of the surface direction is impossible. The best alternative is to approximate the direction of the force as opposing sun position vector.

2.7.3 Typical Acceleration Magnitude: SumbandilaSat

As with aerodynamic drag, we will use the panel with maximum area. For the SumbandilaSat simulation, typical SRP acceleration is found using Equation 2.58:

$$\begin{aligned}\ddot{\mathbf{r}}_{SRP,sumba} &= \frac{-4.51 \times 10^{-6}(0.8 \times 0.65)(1+0.4)}{82} s_{SRP}\mathbf{u}_s\\ &= -4.004 \times 10^{-8} s_{SRP}\mathbf{u}_s \text{ m/s}^2\end{aligned} \tag{2.61}$$

From this it is clear that the magnitude of the solar radiation pressure acceleration is in the range of the order 30 components of the gravity model. SRP only becomes a real issue when the satellite's altitude is higher than 800 km, when the effects of atmospheric drag and gravity perturbations become very small. Only if solar activity is at maximum and/or a large solar sail is deployed, would this have a noteworthy effect on a satellite with an altitude as low as SumbandilaSat.

Chapter 3

SGP4

3.1 Introduction

The most widely used propagation technique is SGP4 [28; 1], a semi-analytical method originally developed in the 1960s. Its popularity is due to it being open source, the regular public release of new TLEs, the relatively small code size and light computational load. Created with the main idea of implementation on computers of the time, this technique uses a set of highly nonlinear analytical equations that will use the least amount of computation effort as possible. It uses a set of mean Kepler elements and a B^* drag term to approximate the average orbit of a satellite. To obtain these, the periodic variations in true satellite states need to be eliminated. SGP4 then needs to try to reconstruct these periodic variations for its output. Greene [1] states that the accuracy of SGP4 is limited by errors in both TLE parameters and internal propagation error intrinsic to the SGP4 force model.

TLEs (Two Line Elements) are mean Kepler element sets compatible with SGP4 and regularly published by NORAD (North American Aerospace Defense Command) for almost every artificial satellite that is still in orbit. These are widely used for smaller satellites [28]. TLEs have several sources of error, and no covariance estimates (quantity that rates the confidence of a TLE being accurate) are published [1].

SGP4 is an analytical method, and inherently it does not produce a set of time differential equations. This causes it to be unsuitable to use as is for an EKF system model (discussed in Chapter 5). It is also not flexible; any alterations to the equations will result in a disharmony, causing system failure.

A SGP4 module, implemented in a C file ready to be compiled in Matlab, was provided by the study leader. This was edited to suit the requirements for simulation. The mathematics behind the SGP4 algorithm is discussed in the Revisiting Spacetrack Report# 3 [29], which also includes a C++ version of SGP4. The CelestTrak website [10] (administered by Dr. Kelso, a co-author of the aforementioned paper) is an excellent source for information regarding SGP4 and TLE's, and also publishes SGP4 source code for C++, Fortran, Java, Matlab and Pascal programming languages as well as for Microsoft Excel.

3.2 History

The launch of Sputnik in 1957 caused space surveillance to become an essential military mission, especially for the United States of America (USA). Since then they have become very efficient at it: by 2004 the USA has catalogued over 27 000 satellites, of which more than 8 000 are still in orbit. The Navy needed a way to prevent fleet units being detected by satellites overhead, and the Air Force wanted to avoid false missile-warnings when a satellite passed the warning system's coverage. Naturally the two pioneering agencies were the Air Force Space Command (AFSPC) and the Naval Space Command (NSC) [30].

To develop such a system, the computer technology limitations at the time was a critical factor. Analytical orbit prediction models require fewer calculations than numerical integration techniques. It was known at the time that special perturbation techniques produced better accuracy than general perturbation techniques, but the sheer amount of satellites in the catalogue left only the analytic propagation techniques viable [30].

3.2.1 First Methods

From the Air Force's side, the National Space Surveillance Control Center (NSSCC) in Bedford, Massachusetts, were the first to formally report on satellite cataloguing procedures in 1959. Satellites were tracked from more than 150 sites, using radars, Baker-Nunn cameras, telescopes, radio receivers and Project Moonwatch. These observations were sent to the NSSCC, where an analyst reduced the data, which was fed to an IBM-709 computer. The updated orbital data was calculated by the computer, and then used by another part of the program to produce geocentric ephemeris. Three products were computed and sent back to the tracking stations for future observing opportunities. These three products were a bulletin, which can be considered as a precursor to the later TLE, and two tables called the General Look Angles Program (GLAP) and the Fence Look Angles Program (FLAP) [30].

The Navy, however, spent most of their effort on automatic satellite detection and cataloguing techniques. In 1958 they began with the Naval Space Surveillance System (NAVSPASUR), commonly known as the Fence. It became operational in 1961. A continuous-wave multi-static radar interferometer was used. It comprised of three transmitters and six receivers along a East-West great-circle arc from San-Diego to Savannah (in the state of Georgia), hence the name the Fence. The raw data, consisting of signal phases and amplitudes, were sent to Dahlgren, where the Navy's only computer able to handle this scope of data flow resided. Using interferometric techniques, this computer converted the data into apparent Direction Matrices (DCM's). Satellites (near Earth) pass over the Fence four to six times per day, and are, on average, detected by four receivers upon each pass. Of the orbits visible to the Fence, more than 98.5% could be updated without any human aid. In 1961 the program was moved to an IBM 7090 computer, which reduced the processing time of a satellite orbit from 15 minutes to just 5. That same year it was reduced even further to about 1 minute, by using new programming and data-handling techniques [30].

3.2.2 Theoretically Founded Methods

1959 was an exciting year for the satellite community. This was the first time that the study of zonal harmonics of Earth's gravity field was incorporated into orbit propagation methods. Both Dirk Brouwer, under Project SPACETRACK, and Yoshide Kozai, published different solutions for this problem on adjacent pages in the Astronomical Journal of that year. These two publications serve as the foundation of most analytical methods that exist today [30] .

The Brouwer method was expanded in 1961 to include basic atmospheric drag effects. At first this inclusion was computationally too expensive to run on the computers of the day (for all the catalogued satellites). After some new atmosphere theory research became available, Max Lane developed a more complete and compact solution in 1965. Lane and Cranford combined Brower's method for its gravitational model, and used a power density function to model the atmosphere. This approach is the essence of SGP4. Another notable contribution was that of R. H. Lyddane, who in 1963 showed that Brouwer's method, based on Delaunay variables, could be reformulated to Poincaré variables. This avoided small divisors such as eccentricity and the sine function of inclination [30; 2].

3.2.3 Implementation

The process to bring this theory into operation took two paths. The first branch was NAVSPASUR, which adopted the Brouwer solution with the Lyddane modifications, which is known today as the Position and Partials as functions of Time (PPT) technique. Richard H. Smith, adopting some simple drag model ideas from King-Hele, implemented the PPT model on an IBM 7090 computer. The drag model was semi-analytical, and assumed that atmospheric drag has a quadratic time effect on a satellite's mean motion. The coefficients were solved for during the process of orbit determination. PPT did however differ from Brouwer's original solution in its definition for mean motion: while Brouwer defined it in the sense of the Keplerian formula, PPT included zonal secular perturbation rate of mean anomaly. This meant that the PPT mean motion is closer to Kozai's method than that of Brouwer. Only much later in 1997, after the development of SGP4, would Lunar, Solar and Earth tesseral gravity components be included to form the current PPT3 model [30].

In 1961 the NSSCC was renamed as the Space Detection and Tracking System (SPADATS) Centre. This is where the second branch of operational implementation started. A Philco Model 211 computer was used to host the program. They used a combination of Brouwer and Kozai's methods, but transformed the parameters to a series of non-singular ones to avoid the small divisors problem. From Brouwer, only the long and short-period position terms that did not include eccentricity factors were used. Kozai's non-Keplerian mean motion to semi-major axis relationship was used. Today this system is known as the Simplified General Perturbations (SGP) model. A drag model similar to that of Smith, except for the assumption that perigee height remains constant, was included [30].

Because analytical density models were outperforming empirical ones, it was decided to implement Lane and Cranford's methods. But because of the number of satellites in the catalogue, a simplified version was implemented and put into operation in 1970 [30]. This modification was the first in a series that led to the modern SGP4 model. SGP4 has received some adaptations over the years. When the first high eccentric 12-hour period satellite was launched in 1965 it was clear that some Lunar and Solar gravitational terms needed to be taken into account. Earth's tesseral gravity effects also started to play a role. Bruce Bowman developed a semi-analytical solution in 1967. In 1977 Dick Hujsak incorporated Bowman's work, which can be considered for the most part the basis of the modern SGP4 version. Despite the name similarity between the original SGP model and SGP4, their mathematical techniques are in fact quite distinct [29].

In 1980 the refined SGP4 source code was publicly released as Spacetrack Report Number 3 (STR#3). Four other orbital propagation models, SGP, SDP4, SGP8 and SDP8, were also released with this report. The SDP (Simplified Deep Space Perturbations) models were intended for deep space orbits, while SGP8 and SDP8 contained corrections for special cases of reentry. All of these models were "generally" compatible with TLE data. Since then, many independent researchers and organisations discovered deficiencies, and many custom changes to the source code have been made. This, along with a STR#6 report which did not enjoy as much circulation as STR#3, left the satellite community with a variety of SGP4 source codes available, and raised doubt as to which is the latest. This lead David Vallado, Paul Crawford, Richard Hujsak and T.S. Kelso to collaborate in developing a unified SGP4 model in 2006 [29].

3.3 Two Line Element Set (TLE)

The classical elements vary due to perturbations. Earth's full gravity model explains small oscillations that occur, while aerodynamic drag has secular effects. The SGP4 propagation technique is initialised with a set of mean classical elements, free of these perturbation's short term effects. This mean set is referred to as the TLE set. NORAD produces TLEs of almost all satellites on a regular basis. A new TLE

set is published when the previous set's propagated trajectory differs more than 5 km (90 % confidence) from the current measurements, but empirical evidence shows otherwise [31]. TLEs can be found on the CelesTrak website [10], which also offers the option to request a satellite's TLE history for a given date interval.

3.3.1 TLE Format

Figure 3.1 shows the format of a typical TLE. Table 3.1 explains the interpretation process to extract the relevant information that SGP4 requires [32; 10].

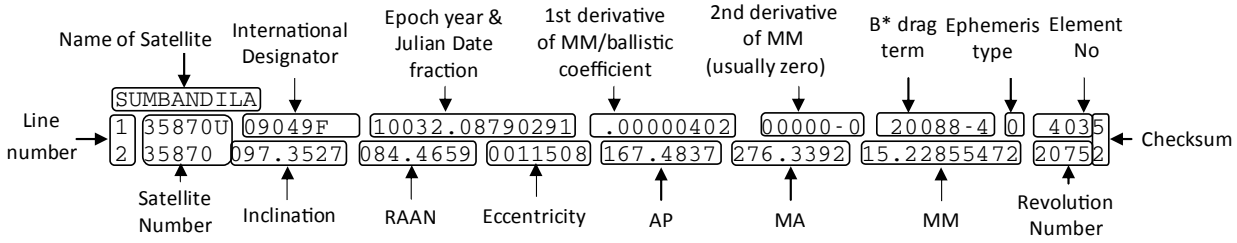


Figure 3.1 – Format of a TLE set.

Table 3.1 – Description of TLE information extraction.

Information	Extraction procedure
Epoch/Timestamp	The first two digits indicate the year, and the trailing part contains the Julian fraction of days since the start of the year. This is explained in detail in Appendix A.1.4.
B*	Also referred to as the radiation pressure coefficient, this is a term that SGP4 uses to determine the drag effect on this particular satellite. It is defined as $B^* = \frac{C_D A}{m} \times \frac{\rho_0}{2}$ (units in (Earth radii)$^{-1}$), and is thus a catch-all term that allows SGP4 to approximate the effect of drag without having knowledge of the atmosphere nor the satellite's ballistic properties. B* describes how susceptible a satellite is to aerodynamic drag: the higher it is, the greater effect drag has on the satellite. A leading decimal must be applied to this term, and the last two digits represent the applicable power of 10 ($B^* = 0.200\,88 \times 10^{-4}$ in Figure 3.1).
Inclination, RAAN, AP, MA	These Kepler elements are simply presented in degrees.
Eccentricity	A leading decimal needs to be applied to obtain the eccentricity ($e = 0.0011508$ in Figure 3.1).
MM	MM is given in number of orbits per day. Note that no trailing space will occur when the revolution number exceeds 9999.

3.3.2 TLE Accuracy

Kahr et al. [31] investigated the accuracy of the NORAD TLEs. TLEs of various ages were used in the SGP4 propagator and the positions were compared to that of on-board GPS receiver measurements. See Subsection 4.1 for a more detailed discussion of the study. The three satellites in this study were CanX-2 (a 3U CubeSat in a Sun-synchronous 635 km altitude orbit with a NovAtel OEM4-G2L GPS receiver

[33]), PROBA 2 (a $600 \times 700 \times 850$ mm, 130 kg satellite in a Sun-synchronous 719 km altitude orbit) and PRISMA Tango (a $800 \times 800 \times 300$ mm, 48 kg satellite at an altitude of 755 km). Figure 3.2 shows the error that the NORAD TLEs produced. The X-axis shows the TLE age; negative values indicate that the TLE was propagated backwards in time. From these plots we can see that the propagation error was at minimum during the 5 days prior to TLE epoch. We can therefore deduct that NORAD most likely uses an arc of the previous 5 days to estimate their TLE parameters. From Figure 3.2 it should be clear that a 1-2 km RMS position error after 24 hours of propagation can be expected when the traditional NORAD TLE and SGP4 combination is used.

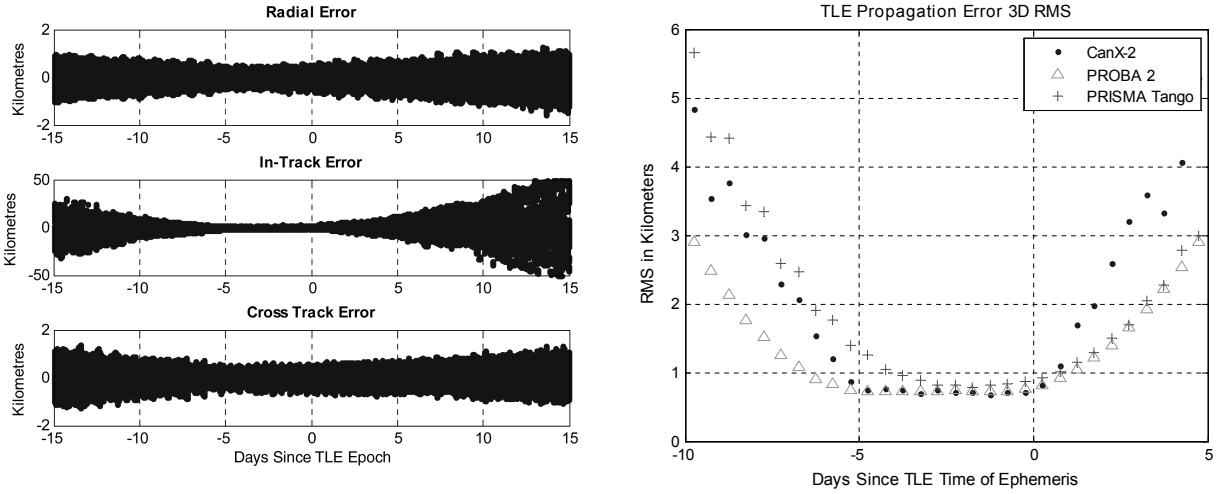


Figure 3.2 – Propagation error of SGP4 used with NORAD TLEs. NTW position error of CanX-2 (left) and RMS absolute position errors of three LEO satellites (right) [31].

3.4 Simulation on SumbandilaSat and NigeriaSat GPS data

3.4.1 First Implementation

SumbandilaSat and NigeriaSat GPS data were used to test the accuracy of the provided SGP4 module. The propagator was fed the latest TLEs prior to the first GPS data timestamp.

The time lapse between the TLE epoch and GPS measurements is required to synchronise the propagator to the measurements. Without a precise synchronisation the in-track error can be enormous. Note that the GPS timestamps are in GPS time format, and leap seconds need to be taken into account. Calculation of the synchronisation time can be found in Equation 3.1. $t_{GPS,0}$ indicates the first GPS measurement's timestamp, t_{TLE} indicates the TLE epoch, and t_{ls} indicates the leap seconds up to the concerning date. Note the use of Algorithms A.2 and A.3 (in Appendix A) to obtain the Julian dates.

$$\begin{aligned} t_{sync,Sumbandila} &= t_{GPS,0} - t_{TLE} - t_{ls} \\ &= (\text{GPS week } 1569 \text{ second } 114032) - (2010 \text{ DoY } 32.08790291) - 15 \text{ s} \\ &= 86400\big(2455228.819814815 - (2455196.5 + 32.08790291)\big) - 15 \text{ s} \\ &= 20022.1886 \text{ s} \end{aligned} \tag{3.1}$$

$$\begin{aligned} t_{sync,NigeriaSat} &= t_{GPS,0} - t_{TLE} - t_{ls} \\ &= (\text{GPS week 1238 second } 505825) - (2003 \text{ DoY } 276.74836650) - 13 \text{ s} \\ &= 86400\big(2452916.354456019 - (2452639.5 + 276.74836650)\big) - 13 \text{ s} \\ &= 9153.1345 \text{ s} \end{aligned} \tag{3.1}$$

Figures 3.3 and 3.4 respectively show how the SGP4 module fairly accurately predicted the SumbandilaSat and NigeriaSat position and velocity. For illustration purposes, it propagated in regular time intervals of 30 seconds from the start of the TLE epoch, and stopped when the first GPS sample was less than 30 seconds away. A propagation was then performed to synchronise the propagator and first GPS measurement. The SGP4 module then propagated in sync with the rest of the GPS measurements.

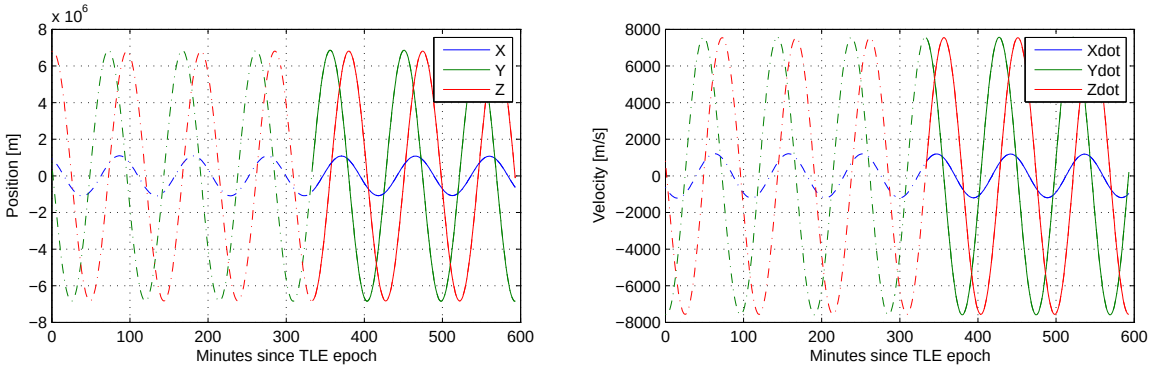


Figure 3.3 – SumbandilaSat propagated states (dashed) compared to GPS data (solid) in the ECI frame.

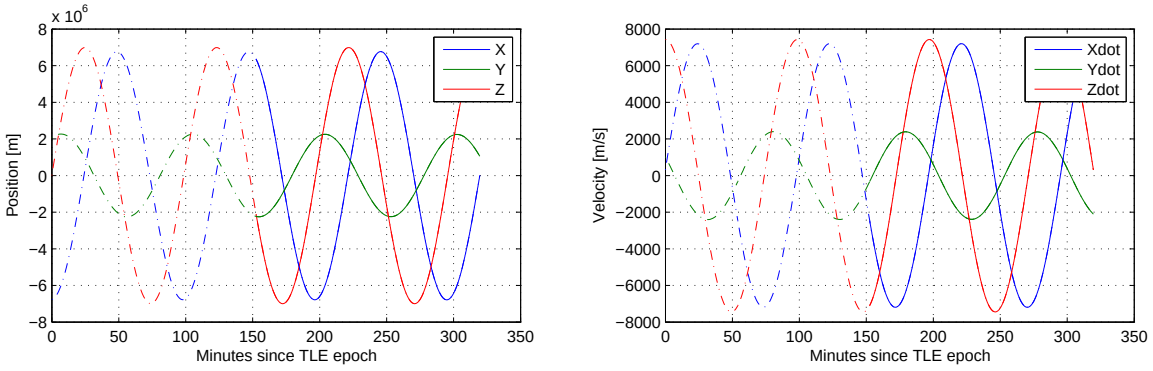


Figure 3.4 – NigeriaSat propagated states (dashed) compared to GPS data (solid) in the ECI frame.

The error is defined as the difference between the GPS measurements and the SGP4 propagated states. This is plotted in Figure 3.5. The larger error of SumbandilaSat is due to it having an older TLE (about 340 minutes) than NigeriaSat (about 150 minutes) to propagate from. The sudden jumps in the error are due to corrupt GPS measurements. The ECI frame does not reveal much about the nature of the SGP4 error. The satellite coordinate frame (defined in Appendix A.3), the NTW frame in particular, provides much more insight into the nature of the position error.

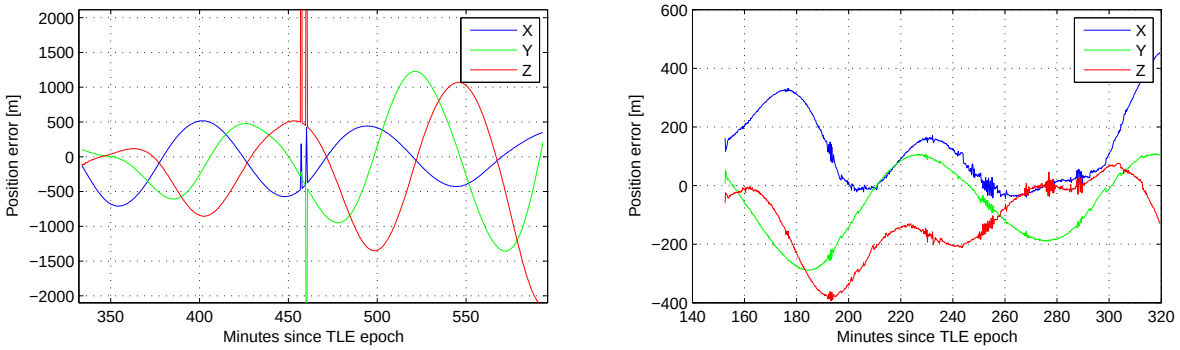


Figure 3.5 – Position error (in the ECI frame) of the SumbandilaSat (left) and NigeriaSat (right) SGP4 propagator.

3.4.2 Satellite Orbital Frame Analysis

We'll use the propagated states to determine the NTW DCM. The new transformed error is found using Equation 3.2, and is plotted in Figure 3.6.

$$\mathbf{r}_{NTW\ error} = \begin{bmatrix} \mathbf{N} \\ \mathbf{T} \\ \mathbf{W} \end{bmatrix}_{SGP4} (\mathbf{r}_{\text{SGP4}} - \mathbf{r}_{\text{GPS}}) \tag{3.2}$$

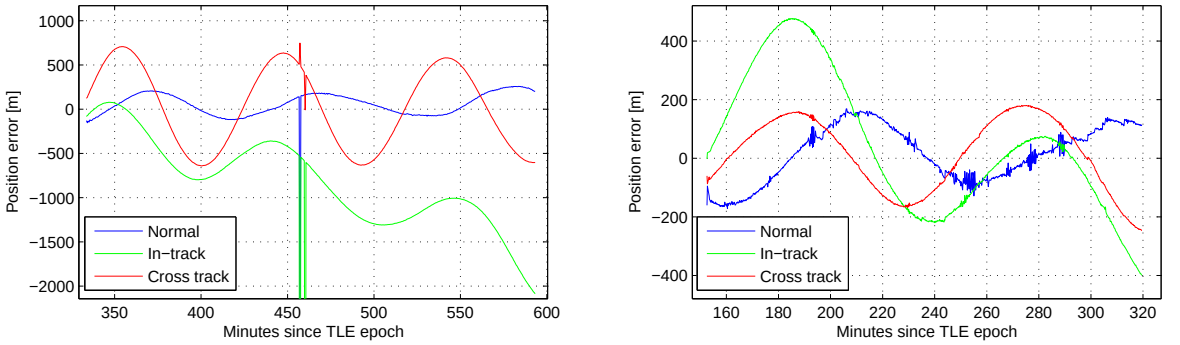


Figure 3.6 – Position error (in the NTW frame) of the SumbandilaSat (left) and NigeriaSat (right) SGP4 propagation.

A clear oscillation with a period of one orbit is observed. This suggests that some of the orbit description parameters in the TLE might be inaccurate. SGP4 might perform better if we introduced Δ parameters to slightly adjust the Kepler elements in the TLE. Apart from oscillation, the in-track error also presents a linear trend.

In-track Error

The in-track component presented the most prominent error. The oscillations are due to incorrect eccentricity or AP parameters. A linear trend is also clearly observed. This is caused by miscalculation of aerodynamic drag. Adjusting the B* parameter in the TLE can resolve this linear effect.

Normal Error

In both cases the normal error is smaller than that of the other errors. As with the in-track error, the slight oscillations can be due to incorrect eccentricity or AP parameters. Upon inspection of the NigeriaSat and SumbandilaSat (close up in Figure 3.7) normal error, we also see a slight offset and linear trend (insignificant compared to the in-track error). The linear trend is due to drag, and the offset due to the accumulation of the drag error since the TLE epoch.

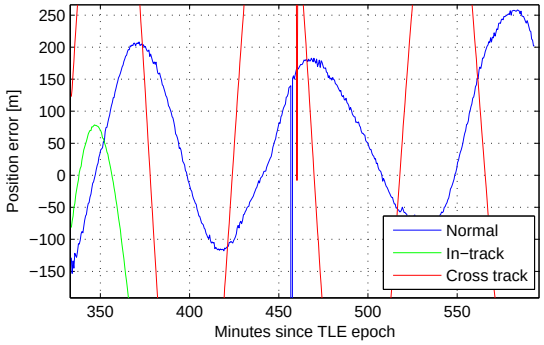


Figure 3.7 – NTW error of SumbandilaSat, focused on normal error.

Cross-track Error

Cross-track errors can be caused by the inclination, RAAN, or incorrect rotation of the GPS measurements (from ECEF to ECI). As the average inclination of a satellite's orbit does not vary much over time, we expect this to be accurately estimated in the TLE. Upon inspection of the SGP4 module, it is found that the RAAN parameter does not affect the propagation process of any parameters. Its sole purpose is to initialise the RAAN at epoch. This means that RAAN and rotational mistakes would produce the same results, as RAAN is just an angle around the Earth polar (Z) axis.

3.5 TLE Age

The age of a TLE has a significant effect on an SGP4 module's performance. A clear example is the decay of in-track accuracy over time. As communication time to LEO satellites is limited and used for multiple tasks, the latest TLE cannot always be uploaded. Thus LEO satellites will regularly rely on older TLEs. As discussed in Subsection 3.3.2, the SGP4 position accuracy typically degrades more than 1 km per day. To test SGP4's performance in such cases, we will initialise the SGP4 module with older TLEs prior to the GPS data, and analyse how the performance degraded.

3.5.1 A Short Retrospect of the Two Satellite's Orbits

Before we can do this, we must first examine how the TLE Kepler elements progressed up to the GPS data. After a propulsion manoeuvre, the NORAD (mean Kepler element set) estimators need a few days to settle onto the new mean set of Kepler elements. The NigeriaSat GPS data was recorded just 5 days after its launch, which means that the first few TLE's will also be inaccurate.

Figure 3.8 shows how the altitude, derived from TLE data, of the two satellites progressed. It is clear that SumbandilaSat underwent a manoeuvre to increase altitude at around day 26. Apart from this, a smooth and steady decrease in altitude is seen. NigeriaSat, on the other hand, was launched on day 271,

and that is why we see the NORAD estimators taking a few days to settle. Propulsion manoeuvres on day 310 and 326 are also clear.

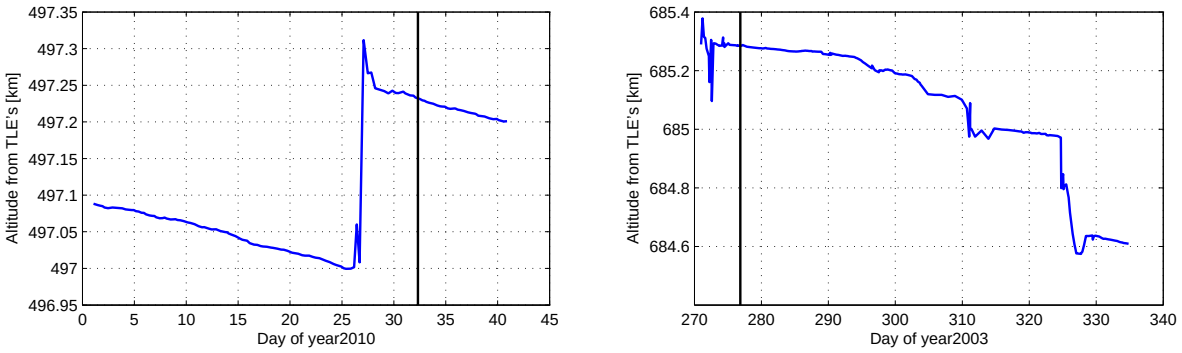


Figure 3.8 – Altitude of SumbandilaSat (left) and NigeriaSat (right) as derived from their TLE history. The GPS data starting point is indicated by the black line.

The satellites' eccentricities are plotted in Figure 3.9. We can see that only the second manoeuvre of NigeriaSat (day 326) was aimed at lowering the eccentricity. The eccentricity estimation takes time to settle after a manoeuvre, even if it was only intended to alter other aspects of orbit (as seen in the SumbandilaSat eccentricity plot after day 26). The uneven progression of TLE eccentricity indicates that the NORAD estimators struggle to determine it when it is this close to zero, and this could be the reason for the oscillations observed in the in-track and normal error. One must also consider the fact that geometric anomalies arise in the Kepler element set theory when the eccentricity approaches zero. Even though these two satellites aren't in exact circular orbits, they are close to it, and this could make the estimation of AP a difficult task.

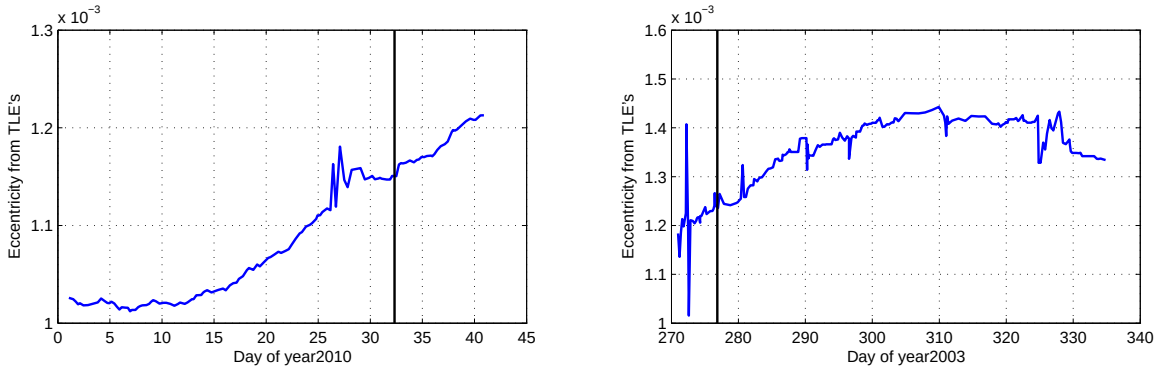


Figure 3.9 – Eccentricity of SumbandilaSat (left) and NigeriaSat (right) from their TLE history.

As expected, the inclination of the two satellites remained constant, while the RAAN and AP progressed linearly with time. Figure 3.10 shows this.

The B^* aerodynamic drag term was also analysed (shown in Figure 3.11). The SumbandilaSat TLEs took about 10 days to settle to the correct B^* value. NigeriaSat, on the other hand, was just launched and the NORAD estimators will still take a while to settle. The strange behaviour of the B^* progression could be due to alternating facets pointing towards the velocity vector during the detumble process. Note that the defective B^* terms in our TLE data will cause orbit decay errors, resulting in major in-track errors.

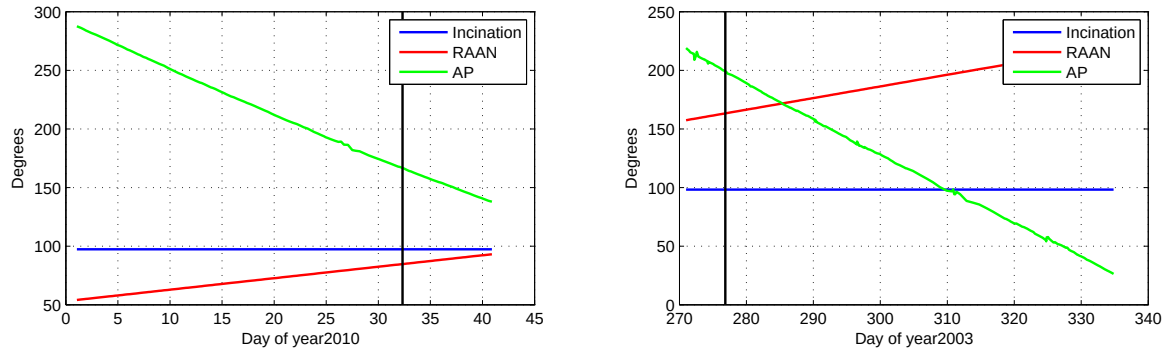


Figure 3.10 – Inclination, RAAN and AP of SumbandilaSat (left) and NigeriaSat (right) from their TLE history.

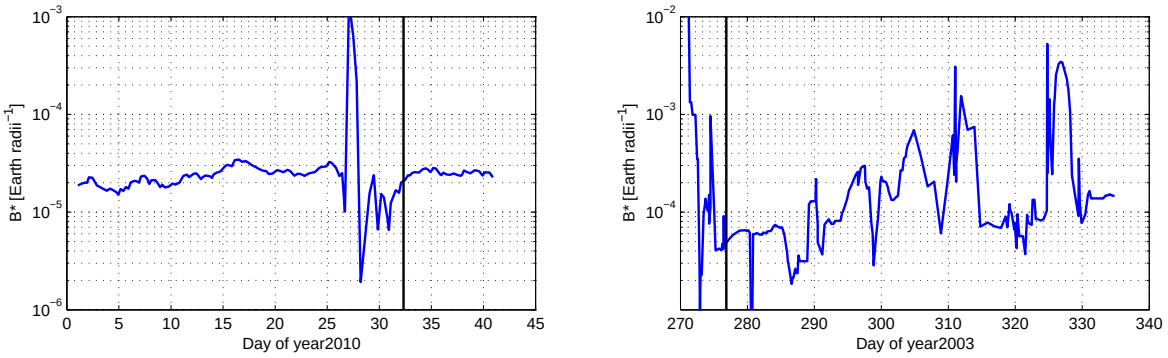


Figure 3.11 – B^* of SumbandilaSat (left) and NigeriaSat (right) from their TLE history.

From SumbandilaSat's eccentricity history it is clear that we cannot use TLEs prior to day 28, as this is where the NORAD eccentricity estimator has settled to the new mean. From its launch, NigeriaSat settles to a stable Kepler element set (by examining both altitude and eccentricity) 4 days prior to the GPS data. In both satellites' cases, the B^* term will most probably be a source of a slight normal and a growing in-track error. Kahr [31] confirms that the drag coefficient estimates are indeed less stable over time. This is due to the difficulty of recognizing the quadratic variation that drag would have on in-track position, as well as the fact that SGP4 assumes a static atmosphere, leaving the B^* estimations to include the time-dependant changes in atmospheric density.

3.5.2 Analysis of Error Growth

To test the degradation of SGP4 accuracy over time, the module was provided TLEs of different age. Figure 3.12 shows the in-track error using TLEs of different age. We see that the older the TLE, the greater the in-track error is. It seems that TLE age does not correlate to the propagated position being ahead or behind the actual position in the orbit track. This is due to the unpredictable B^* error in TLE's just prior to the GPS data.

The normal error, plotted in Figure 3.13, shows little variation until the TLE reached an age of 4 days and 6 hours, when it started to degrade significantly. This is due to the unsettled eccentricity and altitude estimations, as seen in Figures 3.8 and 3.9.

The cross-track accuracy, plotted in Figure 3.14, does not seem to degrade much for older TLEs. The only exception is the NigeriaSat TLE which is 4 days and 6 hours old: it shows a large cross-track

error oscillation. This is most likely caused by a very small errors in the estimated RAAN or inclination parameters (as plotted in Figure 3.10). The scale of the plot was used to accommodate for the secular change over more than 60 days, and subsequently conceals such tiny errors.

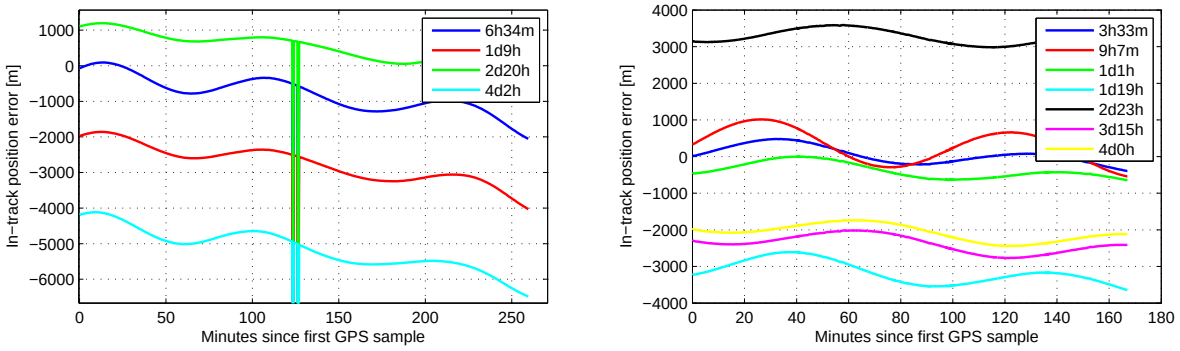


Figure 3.12 – In-track position error of SumbandilaSat (left) and NigeriaSat (right) using TLEs of different ages (see legend for age: d = days, h = hours and m = minutes).

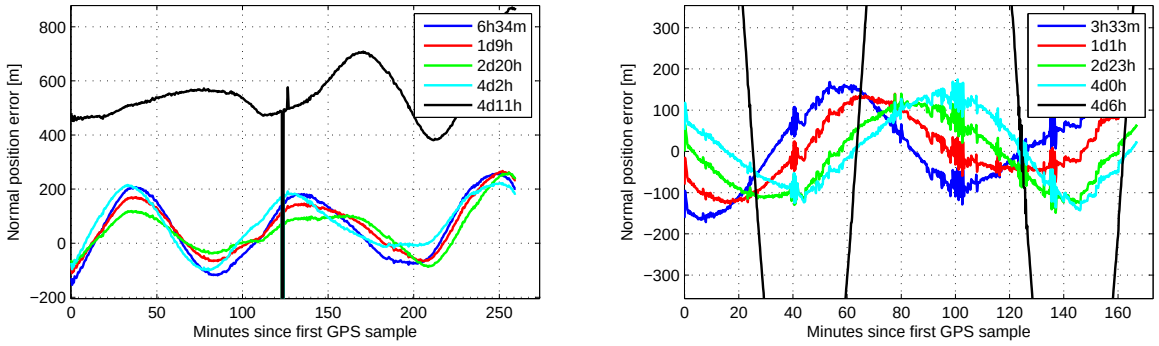


Figure 3.13 – Normal error of SumbandilaSat (left) and NigeriaSat (right) using TLEs of different ages.

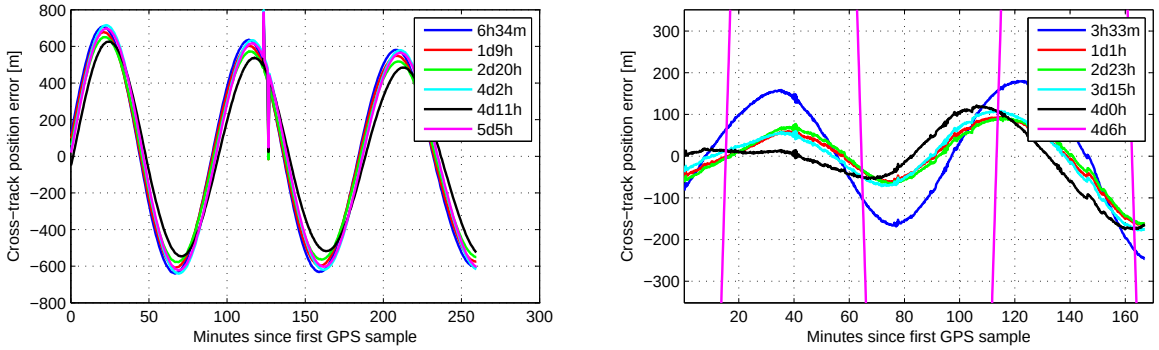


Figure 3.14 – Cross-track error of SumbandilaSat (left) and NigeriaSat (right) using TLEs of different ages.

Chapter 4

Adaptive SGP4

4.1 Previous Studies on GPS-based Enhancement of TLEs and SGP4

Numerous previous attempts at increasing the accuracy of the TLE parameters with GPS measurements have been made. A few of these studies are discussed below.

In 1994 Ernandes developed and copyrighted a shareware software package called VEC2TLE, which has been used in numerous studies, including those by Greene et al. [1] and Kahr et al. [7]. Built for the MSDOS operating system, VEC2TLE takes a vector of position, velocity and time (PVT) states as input and calculates mean Kepler elements. The source code and mathematical algorithms are not available. Andersen [34], upon discussing his own method (for his 1994 Master's Thesis) with Ernandes, noted that they fundamentally use the same principle: Newton's iteration method with numerically calculated partial derivatives and osculating orbital elements as initial guess. Greene used 12 and 20 minute CanX-2 GPS measurement logs of succeeding days as input to the VEC2TLE. The TLE was then fed to the SGP4 propagator, and errors reached 2 km within 5 hours when compared to HPOP (AGI STK's high accuracy numerical propagator [35]) which used the original GPS data. Kahr's research (also based on CanX-2) used only a single GPS measurement and derived an averaged B^* from previous TLEs. Its accuracy matched that of SGP4, and the system was solely intended to hot start the next GPS activation. The VEC2TLE program was downloaded and tested, but as the details of its algorithms are unavailable and the results mentioned above are not accurate enough, we will not investigate its use further.

In 1996 Jochim et al. [36] published a paper which describes a least squares process to determine TLEs independent of the NORAD publications. As OBCs of the time had much more stringent computational load budgets, an analytical propagator method like SGP4 was the only choice, and the system analysed only one GPS measurement every 10 minutes. A sequential estimator was recommended for future use, as the least square method employed required the on-board storage of many previous measurements. The paper states that the research was intended on small satellites, but the CubeSat concept was only introduced three years later, so it was most likely aimed at microsatellites. The goal was to mitigate drawbacks such as data gaps and scatter, as well as the serious noise on velocity measurements, associated with a navigation solution based on GPS (of the time) only. Using 3 hours of GPS data within a 4 day period resulted in a simulated maximum position error of 4 km (equivalent to about 0.5 s timing or 0.5° pointing errors). It must be noted that the simulation used the USA/French TOPEX/POSEIDON satellite, which has an altitude of 1336 km, and subsequently the effect of aerodynamic drag was cut out. This thesis is aimed at the LEO environment, though, where aerodynamic drag is well alive and a major concern.

Montenbruck, a co-author of the aforementioned paper [36], used an epoch state filter for improving the TLE parameters in 2000 [3]. It employed an estimation technique similar to that of the measurement

update of a Kalman filter. This research was based on GPS measurements of the MicroLab-1 satellite and MIR (Russian) space station. The B^* drag term was also estimated, but an along-track uncertainty growth of 5 km per day was still induced by the inaccuracy of the semi-major axis estimation.

Cho et al. [28] published work in 2002 which also used least square estimation for estimating TLE parameters, but based on KOMPSAT-1, a ±500 kg satellite at 685 km altitude [33]. After 7 days the position error was 7 km.

Another study by Kahr et al. [31] implemented a least squares estimation on intermittent GPS data. This study also used the CanX-2 GPS data, but the PROBA 2 and PRISMA Tango satellites, both carrying twin redundant Phoenix GPS receivers, were also included. Using four 10 minute GPS windows daily, 5 and 32 day data arcs yielded 2 km and 5 km in-track errors, respectively, after another five days of propagation. Using less frequent but longer windows (ex. one 40 minute window daily) exhibited very bad results, which is due to worse spread of Earth ground tracks.

Vallado presents a comprehensive mathematical explanation of the OD problem using Differential Correction in his *Fundamentals of Astrodynamics and Applications* textbook [12], and further discussion is made in another paper [37] by him. Although only previous TLEs were used to find the converged new TLE, the algorithms explained are not limited to TLEs as input, and PVT states can also be used as input. The study did not include the results when PVT states were used. ICESat (970 kg, 600 km altitude) was the only LEO satellite considered. Using 1.7 days' TLEs to fit a new TLE, the SGP4 error grew to about 5 km after propagating for two days. Further investigation into this work is suggested (see Chapter 6).

While all of these methods exhibit a major improvement over the regular TLE and SGP4 system, our goal of achieving a 500 m accuracy has not been matched by these studies. We will, however, be able to have a window in at least every orbit, easing the task at hand.

4.2 Δ Parameters

In this section we will alter the SumbandilaSat and NigeriaSat TLE parameters in order to confirm that an improved SGP4 performance can be obtained by enhancing these parameters.

4.2.1 In-track Correction

The in-track position of a satellite is highly dependant on its altitude. This is because altitude determines its velocity (Kepler's 2nd and 3rd laws), and therefore a slight offset in normal position estimation will result into a growing in-track error.

In Figure 3.6 the SGP4 propagator's position prediction seems to be getting more and more behind in the orbit (negative growth of in-track error). Decreasing the altitude (by increasing the mean motion in the TLE) might seem to be the obvious solution to increasing the propagation velocity. We must remember, however, how Figures 3.6 and 3.7 revealed a very slight linear trend in the normal error. This suggests a deficient aerodynamic drag model, as the orbit is not decaying fast enough. Increasing the drag will solve the slight normal error linear trend by slightly decreasing the altitude as time passes. This will, in turn, then also solve the high linearity trend in the in-track error, as satellite velocity increases when lowering altitude.

Atmospheric drag is presented in the TLE by the B^* term. To adjust the estimated impact of drag we will simply multiply B^* before we feed it to the SGP4 module, as shown in Equation 4.1.

$$B^*_{SGP4} = \alpha_{B^*} B^*_{TLE} \tag{4.1}$$

We want this adjustment to mitigate the satellite's linear in-track error growth. Figure 4.1 shows the in-track error when using various values for α_{B^*}. We can see that α_{B^*}'s of 60 for SumbandilaSat and 150 for NigeriaSat significantly diminishes the growth of in-track error.

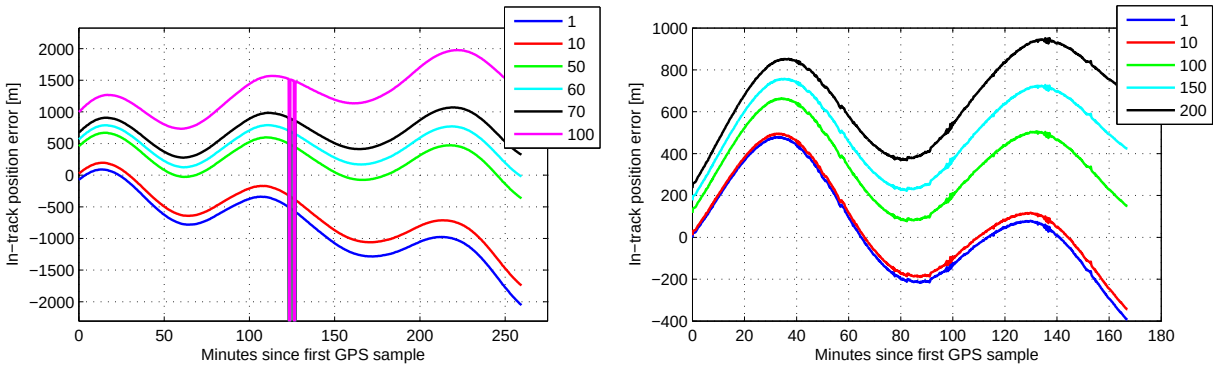


Figure 4.1 – In-track error of SumbandilaSat (left) and NigeriaSat (right) using various α_{B^*}'s (see legend).

Adjusting the drag term also introduced an offset in the in-track error. We can accommodate for this by introducing a time offset. This is done by incorporating a Δt term into Equation 3.1 to obtain Equation 4.2.

$$t_{sync} = t_{GPS,0} - t_{TLE} - t_{ls} + \Delta t \tag{4.2}$$

Table 4.1 shows the approximate Δ parameters for both satellites when using the TLE closest to the GPS samples.

Table 4.1 – Best approximated Δ parameters used to improve the SGP4 in-track error.

	α_{B^*}	Δt
SumbandilaSat	63	-0.068
NigeriaSat	150	-0.064

The new in-track error for both satellites is plotted in Figure 4.2. This is an immense improvement on in-track error when compared to using no Δ parameters.

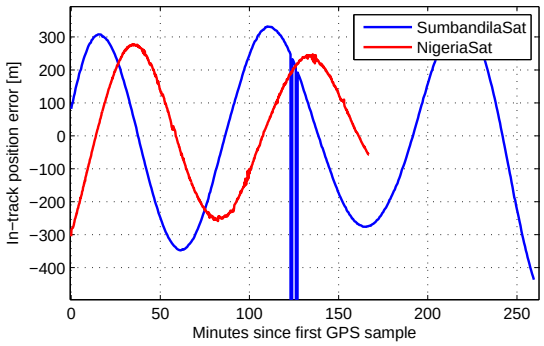


Figure 4.2 – In-track error of SumbandilaSat (blue) and NigeriaSat (red) using approximated optimal α_B and Δt values.

4.2.2 Cross-track Correction

As mentioned before, cross-track error can be accounted for by adjusting inclination and/or RAAN. Instantaneous Kepler elements of the GPS data and SGP4 propagated states can be determined by using the technique described in Sub-subsection 2.3.4. The inclination and RAAN of SumbandilaSat is plotted in Figure 4.3. The noisy appearance of the GPS instantaneous Kepler elements is due to the noise on the velocity measurements. We can see that both the inclination and RAAN have a slight constant offset.

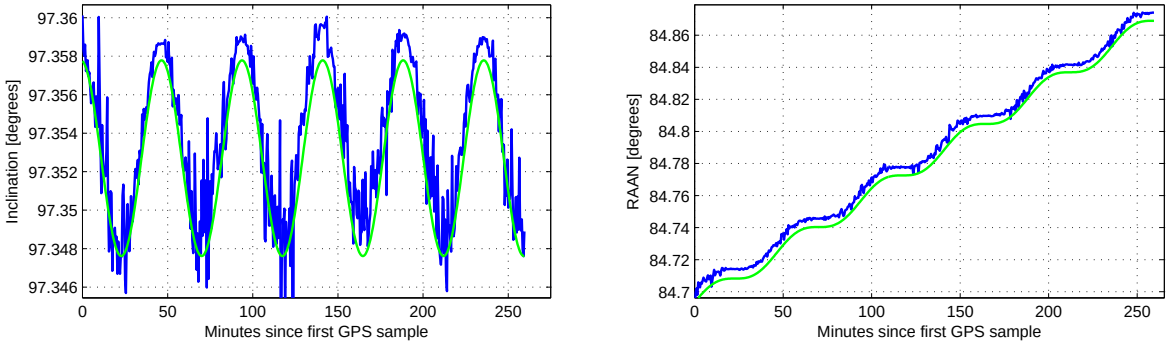


Figure 4.3 – Inclination (left) and RAAN (right) of SumbandilaSat GPS data (blue) and SGP4 propagator (green).

NigeriaSat also presented the same difference. Let's introduce Δi and $\Delta\Omega$ terms, and implement them as shown in Equation 4.3.

$$\begin{aligned} \Omega_{SGP4} &= \Omega_{TLE} + \Delta\Omega \\ i_{SGP4} &= i_{TLE} + \Delta i \end{aligned} \tag{4.3}$$

The approximate best values for Δi and $\Delta\Omega$ for this case is given in Table 4.2. The new cross-track error of both satellites is plotted in Figure 4.4. A significant improvement is observed, especially in SumbandilaSat's case.

Table 4.2 – Best guessed $\Delta\Omega$ and Δi parameters used to improve the SGP4 .

	$\Delta\Omega$	Δi
SumbandilaSat	0.0052	0.0012
NigeriaSat	0.0013	0.0013

4.2.3 Conclusion of Δ Parameters

The results from this section indicate that the NORAD TLEs are indeed flawed and can be improved. We can thus embark on an investigation to use the GPS measurements to refine the TLE parameters.

Although the adjustments made to TLE parameters improved the results, clear oscillations in the errors are still present. These are caused by cyclical disturbance forces not accounted for by SGP4 [1]. Such perturbations include higher order gravity, 3rd body attractions, SRP, and tidal forces. We will not be able to accommodate for these errors as they are inherently due to the SGP4 (and to any analytical method) design.

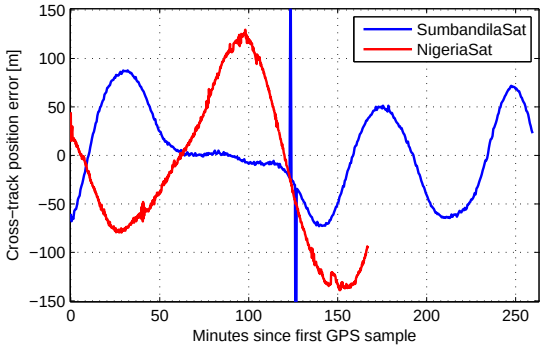


Figure 4.4 – Cross-track error of SumbandilaSat (blue) and NigeriaSat (red) using best approximated $\Delta\Omega$ and Δi values.

These adjustments were also based on a short period of GPS measurement references, and the accuracy will most probably deteriorate quickly after this period. Improper drag modelling and erroneous B^* terms in TLEs were the key reasons for the growing in-track error.

4.3 New Simple aSGP4 Approach

Now that we know that adjusting the SGP4 parameters can significantly improve the SGP4 performance, we will try to automate this process. Even though SGP4 uses mean Kepler elements as input, we will assume that the difference between the GPS measurements' and SGP4's output's instantaneous Kepler elements over a period of time can give a good indication of the mean element set error. Two methods are proposed: a first order IIR filter and simply taking the mean over a period. After the GPS receiver is switched off, the SGP4 algorithm is then reinitialised with the estimated difference added to its TLE derived elements. We will call this system the adaptive SGP4 (aSGP4 for short) technique.

For the simulations done in this sections, a 1s upsampled dataset was generated using the later discussed EKF (see Subsection 5.7.2). The EKF was provided GPS data spanning over 100 minutes, after which numerical integration propagation was done to generate a 24 hour dataset. The faster 1 Hz sampling rate was required to obtain a good estimate of the average elements. As this fast rate is not required for SGP4 propagation, a 30 s sample time was used during the phases when the GPS was turned off. As explained in Subsection 5.7.2, noise was added to the generated GPS measurements. When calculating the aSGP4 propagation errors, the original generated data (without the noise) was used as the true states.

Figure 4.5 illustrates the setup for the simulation environment. The components will be explained in the rest of this Section.

4.3.1 Estimating Δ Parameters with IIR Filter

A concise explanation of IIR filter principles can be found in Appendix C.3. The first correction started after 30 minutes. The GPS was turned on every 95 minutes, and the IIR filters were initialised to their respective instantaneous difference. The GPS was activated for only 1 minute every orbit.

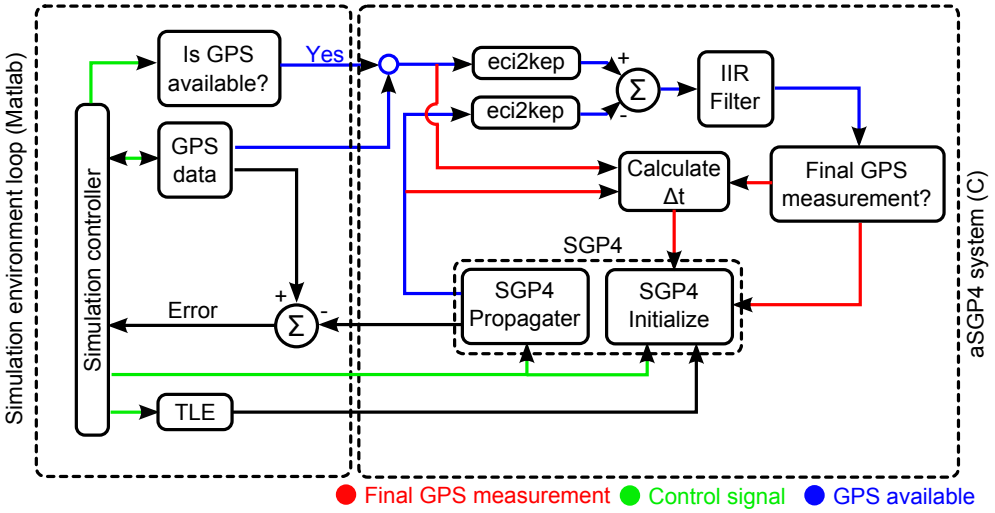


Figure 4.5 – Simulation setup of the aSGP4 system.

4.3.1.1 Orbit Plane Correction

We will first attempt to correct the cross-track error by correcting inclination and RAAN. The following filters were used:

$$\begin{aligned} \Delta i_{k+1} &= \alpha_i \tilde{i}_{k+1} + (1-\alpha_i)\Delta i_k \\ \Delta\Omega_{k+1} &= \alpha_\Omega \widetilde{\Omega}_{k+1} + (1-\alpha_\Omega)\Delta\Omega_k \\ \alpha_i &= \alpha_\Omega = 0.001 \end{aligned} \tag{4.4}$$

where $\tilde{i}$ and $\widetilde{\Omega}$ denote the difference between the SGP4 output and GPS data instantaneous inclination and RAAN, respectively. All tilde parameters (such as $\tilde{i}$ and $\widetilde{\Omega}$) were obtained using the method of Subsection 2.3.4:

$$\begin{bmatrix} \tilde{e}_k & \tilde{a}_k & \tilde{i}_k & \widetilde{\Omega}_k & \widetilde{\omega}_k & \tilde{v}_k \end{bmatrix}^T = \text{eci2kep}(\mathbf{x}_{k,GPS}) - \text{eci2kep}(\mathbf{x}_{k,SGP4}) \tag{4.5}$$

An improvement of instantaneous inclination and RAAN error is clearly seen in Figure 4.6. The RMS cross-track error improved from 530.7 m to 79.78 m.

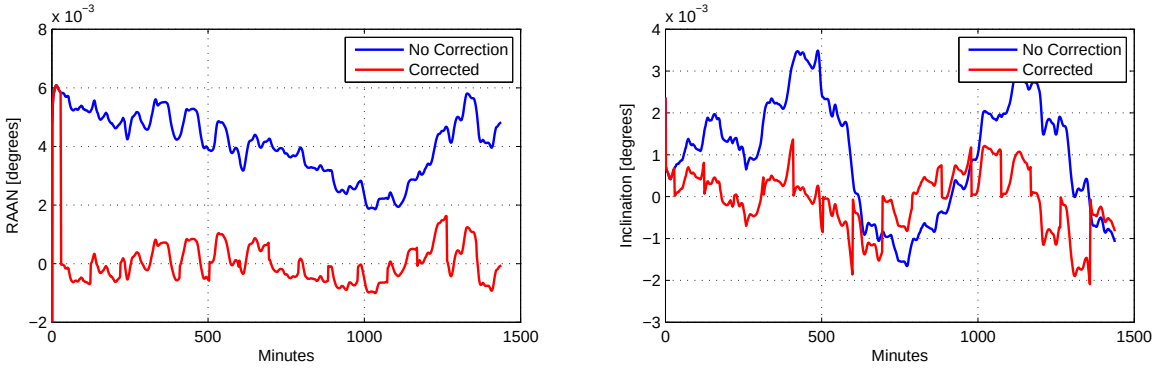


Figure 4.6 – Instantaneous inclination and RAAN errors of aSGP4 compared to the standard SGP4.

4.3.1.2 Orbit Track Correction

As satellite altitude determines orbiting velocity, the in-track error is linked to the normal error. We will assume that altitude (mean motion) is estimated adequately and needs no correction. For orbit description it is the eccentricity and AP that requires attention, and a MA (mean anomaly) adjustment to correct the in-track error. Applying the same IIR principles as previously, with

$$\begin{aligned} \Delta e_{k+1} &= \alpha_e \tilde{e}_{k+1} + (1-\alpha_e)\Delta e_k \\ \Delta\omega_{k+1} &= \alpha_\omega \widetilde{\omega}_{k+1} + (1-\alpha_\omega)\Delta\omega_k \\ \alpha_e &= 0.003 \qquad \alpha_\omega = 0.001 \end{aligned} \tag{4.6}$$

we obtained the instantaneous eccentricity and AP differences shown in Figure 4.7. It is important to remember that MA is defined relative to the AP. Thus changing AP will move the estimated satellite position in its track. To account for this, we define and implement a $\Delta\text{MA} = -\Delta\omega$. As we have an eccentric orbit, this will only really be precise if we worked with TA (true anomaly) instead of MA. Our SGP4 module, however, is initialised by MA, and thus we will have to work with MA. The fact that our orbit is only slightly eccentric marginally relieves this problem, but a final Δt parameter, discussed later, will ultimately resolve this matter.

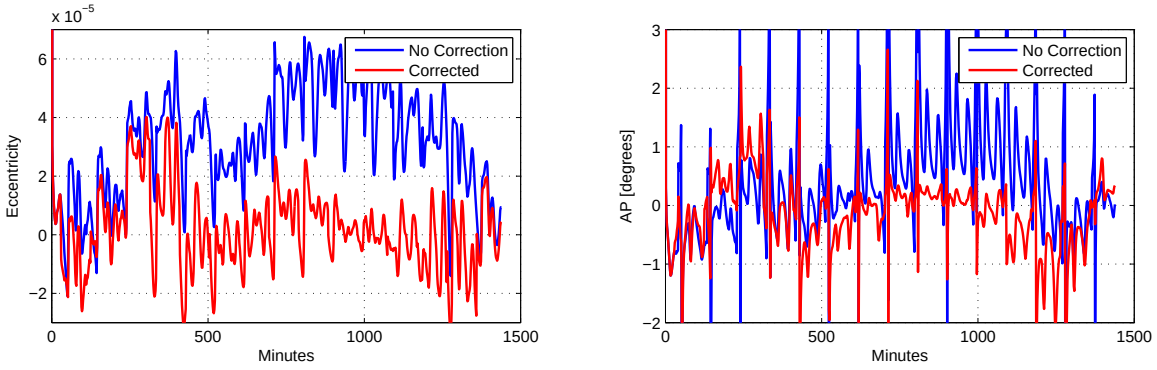


Figure 4.7 – Instantaneous eccentricity and AP errors of aSGP4 compared to the standard SGP4.

The instantaneous Kepler element error improvements of the aSGP4 are summarised in Table 4.3.

Table 4.3 – RMS error of instantaneous Kepler elements for aSGP4 and standard SGP4.

	RAAN [deg]	Inclination [deg]	Eccentricity	AP [deg]
SGP4	4.249×10^{-3}	1.684×10^{-3}	3.921×10^{-5}	1.275
aSGP4	1.093×10^{-3}	0.714×10^{-3}	1.945×10^{-5}	0.763

4.3.1.3 In-track Correction

While correcting eccentricity and AP improves SGP4's orbit track estimation, the position within this track still needs refinement. We devise a Δt parameter, defined as

$$\Delta t_{k+1} = \Delta t_k - \frac{r_{k,\ in-track\ error}}{v_k} \tag{4.7}$$

which will be evaluated after the other Δ parameters have been implemented and the SGP4 module has re-estimated the position of the last GPS sample. Note that although the index parameter is also indicated

by k here, this correction is only performed once just before the GPS is turned off. $\Delta t_0 = 0$ and v_k is the satellite's velocity magnitude at the final GPS measurement. The 24 hour simulation described above was repeated with this Δt concept included. The RMS normal error improved from 216.4 m to 103.02 m, and the in-track error from 3.106 km to 377.12 m.

The Δt parameter proposed above will try to eliminate the in-track error upon the last GPS sample. If we assume that the next in-track error will most likely be in the same direction as previously, we can "overcompensate" for the in-track error. Let's redefine our formula for Δt:

$$\Delta t_{k+1} = \Delta t_k - \alpha_t \frac{r_{k,\ in-track\ error}}{v_k} \tag{4.8}$$

By adjusting α_t we can now modify the in-track correction scheme's aggressiveness. An $\alpha_t < 1$ will create a smoother but slower transition, while $\alpha_t > 1$ will overcompensate the correction, but might yield an unstable system. This design shows a resemblance to an integrator with a α_t gain. A range of α_t values were put in simulation. As expected, the normal and cross-track errors did not show much variation (< 0.2 m) when α_t was changed. An empirically derived $\alpha_t = 1.6$ yielded the best in-track performance for the current setup.

The NTW error of the aSGP4 (with $\alpha_t = 1.6$) compared to that of the normal SGP4 is shown in Figure 4.8. The aSGP4 yielded an RMS 3D error of 305.5 m.

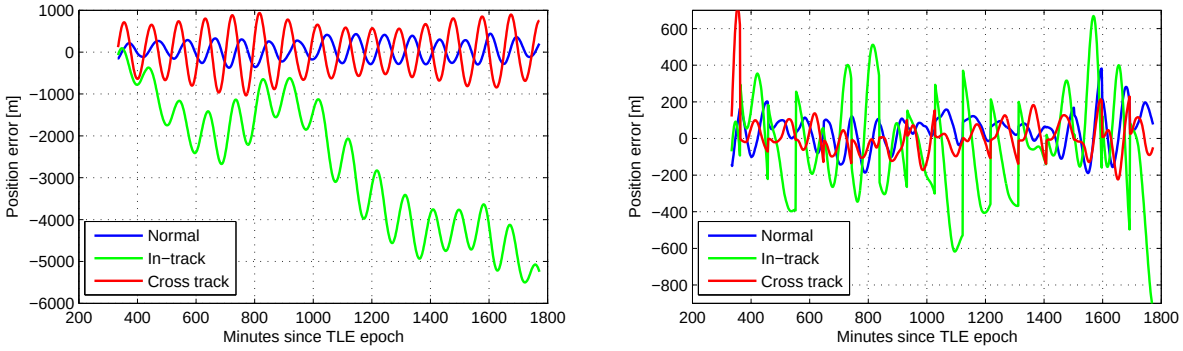


Figure 4.8 – NTW errors of normal SGP4 (left) and aSGP4 using $\alpha_t = 1.6$ (right).

4.3.2 Estimating Δ Parameters with Mean Values

We will now repeat the simulation of the previous subsection, but instead of using IIR filters, the Δ parameters will be calculated by simply taking the mean difference of the instantaneous Kepler elements:

$$\begin{bmatrix} \Delta e & \Delta i & \Delta\Omega & \Delta\omega \end{bmatrix}^T = \frac{1}{n} \sum_{k=0}^{n-1} \begin{bmatrix} \tilde{e}_k & \tilde{i}_k & \tilde{\Omega}_k & \tilde{\omega}_k \end{bmatrix}^T \tag{4.9}$$

where n is the number of measurements (60 in this case as $t_s = 1$ s and $t_{act} = 1$ minute). This method requires less computations than the IIR filter (which has an extra α multiplication).

Figure 4.9 shows the NTW position error of this method in the 24 hour simulation ($\alpha_t = 1.6$). The performance is very similar to that of the IIR filter.

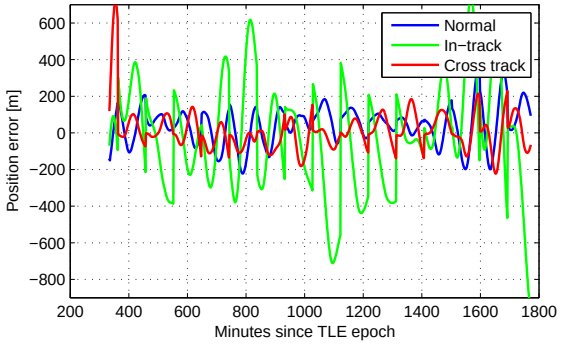


Figure 4.9 – NTW error of aSGP4 (using mean for Δ parameters).

4.4 Optimising aSGP4 Parameters

Until now, the GPS receiver was activated every 95 minutes. This interval was chosen purely because it roughly resembles SumbandilaSat's orbit period of 94.56 minutes. The 1 minute GPS active period was also chosen as a starting point. These parameters have to be optimised, and thus we ran simulations with various α_t, t_{act} and t_{int} parameters. Contour plots of the 3D RMS position errors are shown in Figure 4.10, and the maximum errors in Figure 4.11. From these simulations it is clear that this system will not meet the requirements as set out in Chapter 1 (Table 1.2).

It seems that generally the behaviour is more predictable for a lower α_t, but overall performance decreases when $\alpha_t < 0.8$. From the RMS error the most viable combination seems to be $t_{act} = 4$ and $t_{int} = 40$ with α_t either 0.8 or 1.2. From the maximum error, we see that $\alpha_t = 0.8$ performed better. RMS and maximum errors of 200 m and 1000 m, respectively, can be expected.

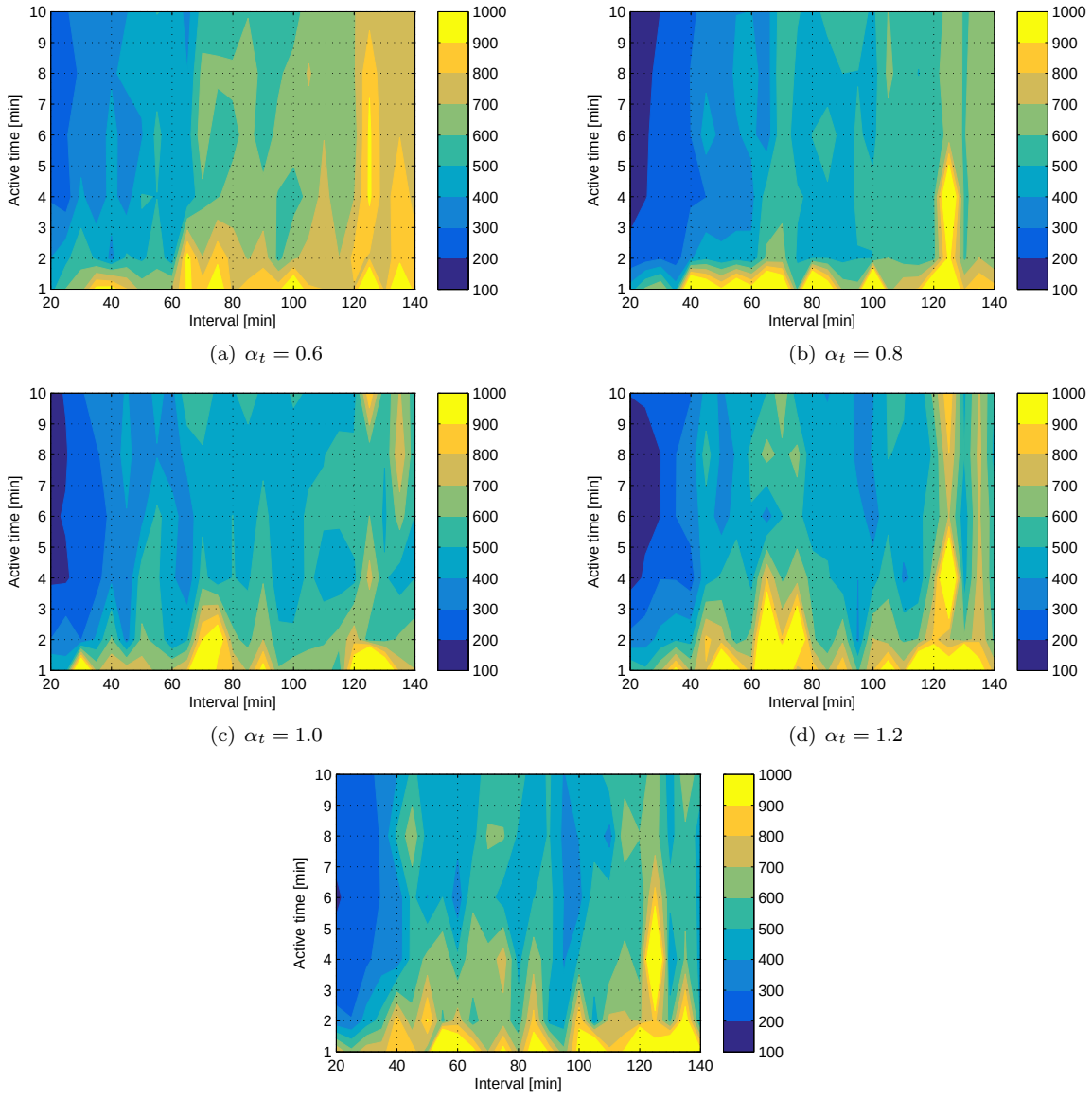
4.5 Power and Computational Budget of aSGP4

From Equation 1.2 and a TTFF range of 1-2 minutes (see Appendix B), we can expect the aSGP4 power consumption factor to be in the range of:

$$\eta_{P,aSGP4} = \frac{t_{act} + \text{TTFF}}{t_{int}} = \begin{cases} \frac{4+1}{40} = 0.125 & \text{minimum} \\ \frac{4+2}{40} = 0.15 & \text{maximum} \end{cases} \tag{4.10}$$

This means that the aSGP4 method will consume about 12.5 % to 15% of a permanently activated GPS receiver's power.

This method requires relatively few computations, but we had to use a high sample rate during the GPS active phase for acceptable results. Using Matlab Profiler on 24 hour simulations we obtained a 16 ms average execution time for the original SGP4 propagator (30 s sample rate). The aSGP4 system, with sampling rates of 1 s and 30 s during GPS active and inactive periods required 66 ms in total to execute. We can approximate the computational load factor using Equation 1.3 as $\eta_{comp,aSGP4} \approx \frac{66}{16} = 4.125$ to complete.



(a) $\alpha_t = 0.6$

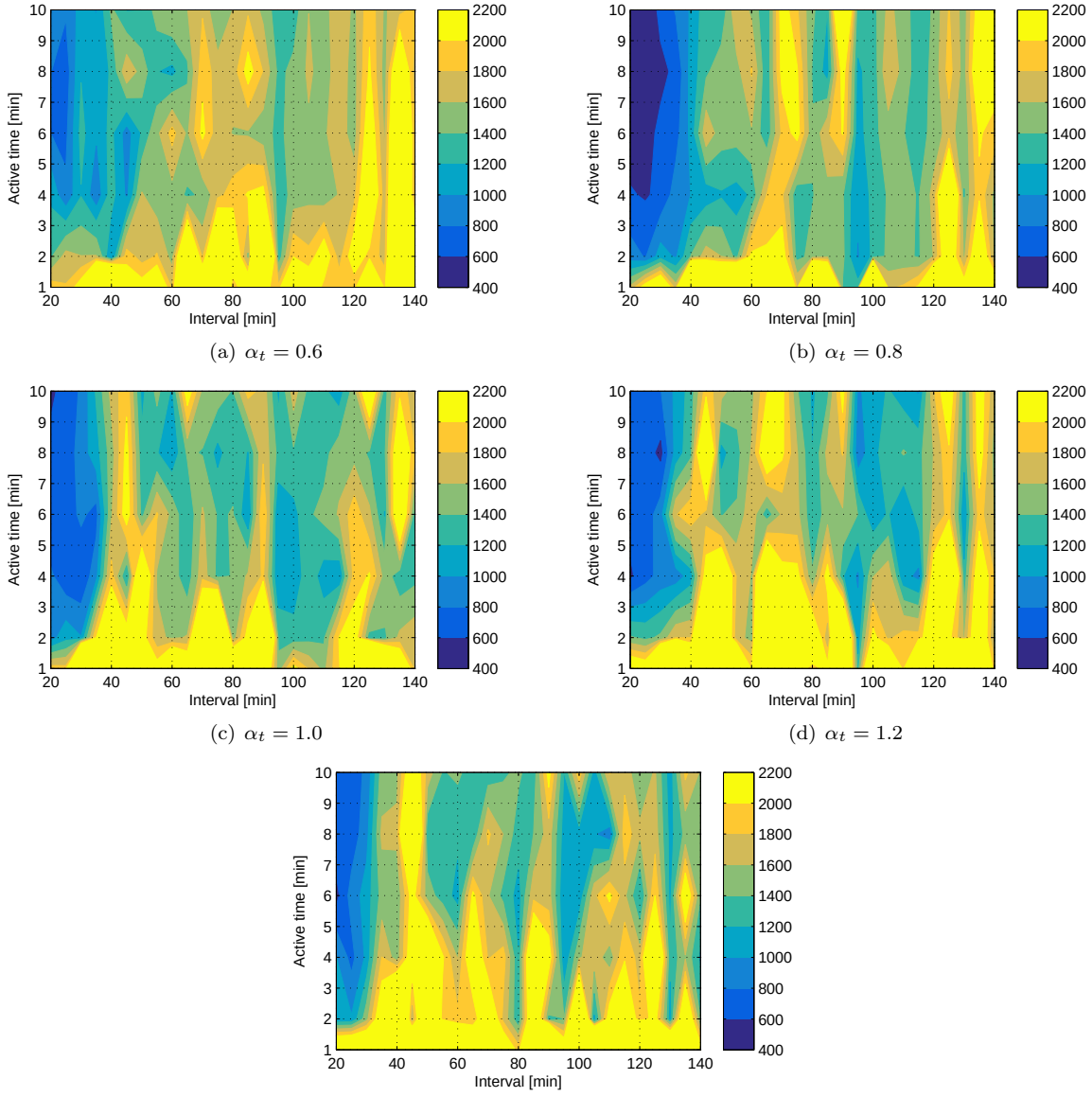
(b) $\alpha_t = 0.8$

(c) $\alpha_t = 1.0$

(d) $\alpha_t = 1.2$

(e) $\alpha_t = 1.4$

Figure 4.10 – RMS 3D position error of aSGP4 for different α_t's.



(a) $\alpha_t = 0.6$

(b) $\alpha_t = 0.8$

(c) $\alpha_t = 1.0$

(d) $\alpha_t = 1.2$

(e) $\alpha_t = 1.4$

Figure 4.11 – Maximum 3D position error of aSGP4 for different values of α_t.

Chapter 5

Numerical Methods and the Extended Kalman Filter

5.1 Previous Studies on Precise Orbit Determination

Extremely precise satellite orbit determination has been around since the 1990s. A notable mission which is frequently mentioned in the literature is the TOPEX/POSEIDON radar altimeter satellite (launched in 1992). Upon its launch there was budgeted for a 13 cm RMS position accuracy. A Kalman Filter approach for achieving this was proposed by Yunck [38] in 1990. Rapid technological advances (enhanced and expanded gravity models along with improved laser ranging and Doppler tracking systems) led to a 3 to 4 cm RMS accuracy by 1994 [39], and later even a 2 cm RMS could be achieved [40]. Jason-1 (launched in 2001), a follow up mission, demonstrated a 1 cm position accuracy [41]. Both of these satellites had dual-frequency GPS receivers on-board. These satellites are orbiting at a 1300 km altitude, and are thus sheltered from the adverse effects of aerodynamic drag.

The CHAllenging Minisatellite Payload (CHAMP), launched in 2000, is another mission which features sub 10 cm position accuracy [42]. This satellite was in an 450 km orbit, and was the first to achieve this accuracy at an altitude this low. Another noteworthy mission is the Gravity Recovery And Climate Experiment (GRACE), which was launched in 2002. It consists of two nearly identical satellites at a separation of approximately 220 km in a 500 km orbit [43]. Both of these satellites were intended for (amongst other tasks) estimating the geopotential (which necessitates such accurate navigation solutions and the low altitude). In addition to this, the GRACE mission was also used to study the temporal variations in Earth's gravity field. Both satellites carry a dual frequency BlackJack GPS receiver developed by the Jet Propulsion Laboratory (JPL). The navigation solution data of these missions could not be obtained for this thesis, but many studies on precise orbit determination has used this data:

- Kuang [42] used the method by Yunck [38] for precise orbit determination of CHAMP. A 70×70 JGM-3 Earth gravity field, atmospheric drag, SRP, Earth radiation pressure and relativity accelerations are included in the force model. Along with the satellite position and velocity states, a drag coefficient and a radiation pressure coefficient is also estimated. A 1σ position accuracy of 47.5 cm was achieved.
- In a study by Van Helleputte [43], both satellites' on-board highly sensitive accelerometers were used to replace the aerodynamic drag and SRP force models. On days of high solar activity, this method was superior to methods that used force models for these two non-conservative perturbations. A 3.4 cm 3D RMS position error was achieved with a least-squares fit and accelerometer data that were processed by JPL. Kang [44] used similar methods for both the GRACE and CHAMP missions.

- Apart from the studies above, many others such as Reigber [45] and König [46] also used the precise GPS orbit data to estimate and refine the geopotential in order to obtain extremely precise positional fits.
- Bock [47] investigated the effect of using only single frequency GPS signals (subsequently exposing the navigation solution to ionospheric delay uncertainties). They still achieved a 10 cm 3D position solution, but had to use a sampling rate of 10 s (which is much higher than what the previous studies have used). The ionospheric delay is only of concern when an accuracy of this degree is required.

All of these studies were, however, based on GPS and/or accelerometer data that was post-processed by JPL before simulation. For example, the GPS satellite reference orbits, from which CHAMP and GRACE navigations solutions were calculated in their simulations, were corrected using ground observations that post-dated the periods simulated. The studies also used powerful computers to obtain the estimates. Thus the results obtained in these studies are unrealistic for our on-board, real-time, computationally constrained problem.

An interesting read is that of Choi [48], which discusses the implementation of an Unscented Kalman Filter (UKF) to the OD problem. Their force model included a 40×40 geopotential, the Sun and the Moon gravity, SRP and US standard atmosphere aerodynamic drag perturbations. A $7(8)^{\text{th}}$ order Runge-Kutta integration method with a 30 s step size was used. Using only the C/A code pseudorange (i.e. single frequency), an 3D position error of 12.1 m RMS was achieved for the CHAMP satellite. When the GPS navigation solutions were used, the position error increased to 13.8 m. The larger error compared to the aforementioned studies is due to this study using the raw (non-post-processed) GPS data for their simulations. The paper states that the solution is intended for on-board application: the program was implemented on the micro-processor used for the spaceborne GPS receiver module, which is a 32-bit 80 MHz Digital Signal Processor. It is intended to host our solution on CubeComputer, which has a 32-bit 48 MHz MCU [11], and must also attend many other duties. Subsequently, Choi's processing power will not be available for this thesis.

In all of the studies above, the GPS data have been continuous and readily available. A situation much closer to ours is investigated by Mander [49], where only intermittent CHAMP GPS data was made available to the estimator. Again, the reduced dynamic method by Yunck [38] was utilised. By processing double and single frequency solutions in a real time situation (non-preprocessed data used), with GPS measurements of 10 minutes in every 90 minutes, they obtained 22 m and 100 m RMS position errors. A Harris-Priester 71 atmospheric density, the Sun and the Moon gravitational forces and cannonball SRP models were used. What is concerning though is the fact that a 100×100 EIGEN-CHAMP03S geopotential model was used. Not only will this consume a great amount of computational power, but this very model is derived from using only the CHAMP precise orbital data [50], which might be biased to fit the specific CHAMP orbit.

5.2 Principles of the Extended Kalman Filter

This section is loosely based on concepts described in the Advanced Automation 813 Course Notes [51] for its intuitive notation. There exists a great wealth of EKF information in the literature; consult Thrun [52], Ribeiro [53] or Labbe [54] for comprehensive discussions. A Kalman Filter is a sequential least squares estimation technique that, given noisy measurements and a faulty model of the system, predicts a linear system's near future state. The Extended Kalman Filter handles nonlinear systems by linearising them around the current estimated states. Internal states are defined and used to describe the system dynamics through interrelating equations. This is called the model. Two updates are used to estimate future

states: a propagation update, which is based on the model, and a measurements update. The expected system noise (errors of modelling the system dynamics) and measurement noise (irregular fluctuations in measurements), are used to weigh the significance of the two updates relative to each other. This is done through a statistical process that will yield an optimal solution (for the given speculated weights).

5.2.1 Defining the Dynamics Model

For our satellite orbit EKF problem, the state $\mathbf{x}$ is defined as the position and velocity Cartesian vectors in the ECI frame. If one wants to estimate SRP and drag coefficients (as done in some the studies discussed in Section 5.1), these would require an additional state each.

$$\mathbf{x} = \begin{bmatrix} x & y & z & \dot{x} & \dot{y} & \dot{z} \end{bmatrix}^T \tag{5.1}$$

Equation 5.2 describes the continuous nonlinear equations for the system dynamics. Note that our model assumes that no external input is given to the system, i.e. that the satellite has not underwent any propulsion manoeuvres. In this case, a propagation update can be performed from the satellite's current state alone. $\mathbf{W}_t$ and $\mathbf{V}_t$ indicate the system and measurement noises, respectively.

$$\begin{aligned} \dot{\mathbf{x}}_t &= \mathbf{f}(\mathbf{x}_t) + \mathbf{W}_t \\ \mathbf{y}_t &= \mathbf{h}(\mathbf{x}_t) + \mathbf{V}_t \end{aligned} \tag{5.2}$$

5.2.2 Algorithm and Implementation

The EKF algorithm used for this thesis is summarised in Algorithm 5.1. Figure 5.1 illustrates the simulation setup. RK4 is a numerical integration method discussed in Section 5.3, the $\mathbf{F}$ (state transition) and $\mathbf{H}$ (observation) matrices are discussed in Section 5.4, and the $\mathbf{Q}$ and $\mathbf{R}$ matrices are discussed in Section 5.5.

Algorithm 5.1 Extended Kalman Filter

initialisation:
Populate $\mathbf{Q}$ and $\mathbf{R}$
$\mathbf{P}_0^+ = \mathbf{I}$
while simulation not finished **do**
  Propagation update:
    determine $\mathbf{F}_k$ using $\mathbf{x}_{k-1}^+$
    $\mathbf{P}_k^- = \mathbf{F}_k \mathbf{P}_k^+ \mathbf{F}_k^T + \mathbf{Q}$
    $\mathbf{x}_k^- = \text{RK4}(\mathbf{x}_{k-1}^+)$
  Measurement update (only if measurements are available):
    $\mathbf{L}_k = \mathbf{P}_k^- \mathbf{H}_k^T (\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R})^{-1} = \mathbf{P}_k^- (\mathbf{P}_k^- + \mathbf{R})^{-1}$
    $\mathbf{P}_k^+ = (\mathbf{I} - \mathbf{L}_k \mathbf{H}_k)\mathbf{P}_k^- = (\mathbf{I} - \mathbf{L}_k)\mathbf{P}_k^-$
    $\mathbf{x}_k^+ = \mathbf{x}_k^- + \mathbf{L}_k(\mathbf{y}_k - \mathbf{h}(\mathbf{x}_k^-)) = \mathbf{x}_k^- + \mathbf{L}_k(\mathbf{y}_k - \mathbf{x}_k^-)$
end while

5.3 Propagation

Traditionally, state estimation EKFs use $\mathbf{x}_k^- = \mathbf{f}(\mathbf{x}_{k-1}^+)$ for their propagation update. As orbit dynamics are highly nonlinear, our step size is relatively large and our measurements will be regularly interrupted for long periods, a more accurate predicting strategy must be resolved. A numerical integration technique is thus required.

Numerical integration techniques have quite a notable computation load compared to analytical orbiting solutions. It is only in the last couple of decades that the technology advancements have made these

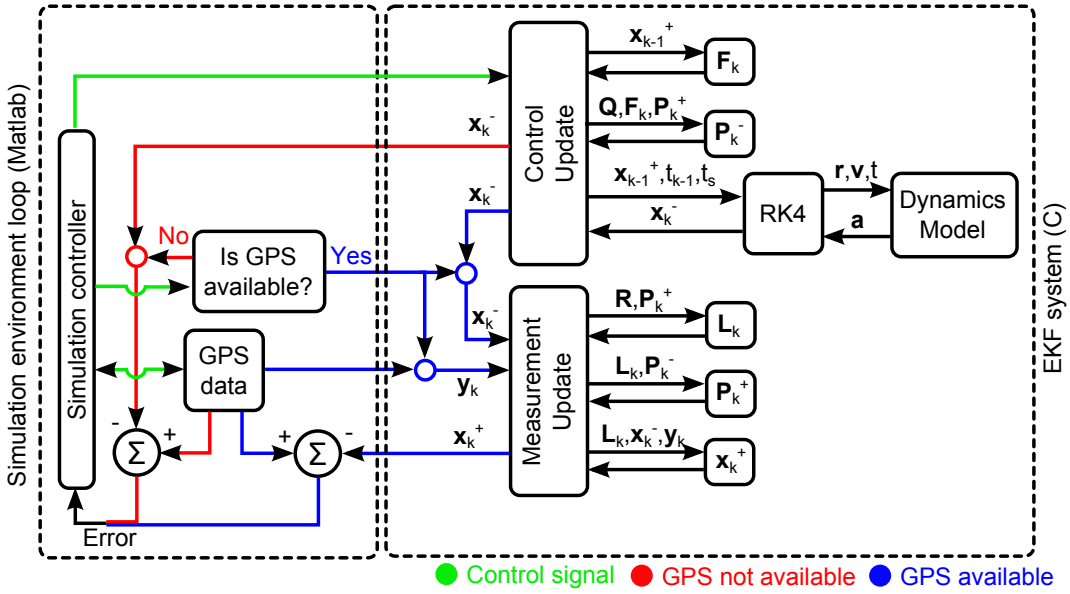


Figure 5.1 – Illustration of EKF system simulation setup.

viable onboard orbit propagation solutions. With the exact state of the satellite and a perfect dynamic model, numerical integration techniques can produce extremely accurate navigation solutions. However, our dynamic model is not perfect (due uncertainties in the atmosphere and truncation of our gravity model), and we cannot use the GPS measurements directly to obtain the exact state of the satellite (as they are noisy). This is why numerical integration techniques mostly go hand-in-hand with some estimation technique for real life applications.

5.3.1 Choosing a Numerical Integrator

Montenbruck [13] recommends three numerical integration methods for precise orbit determination:

- *Runge-Kutta methods*, which are fairly uncomplicated and can be applied to a wide range of problems.
- *Multistep methods*, which are very efficient, but they require the storage of previous data points. They can become less efficient when step-size needs to be varied.
- *Extrapolation methods*, which are very accurate, but more applicable to post-processing applications.

The Runge-Kutta methods are popular, simple and can yield relatively accurate results. All the accelerations acting onto the satellite are modelled and described as a function of position, velocity and time. This function is evaluated for different stages during the step-size interval, and then combined to predict the satellite's state in the near future.

Higher order Runge-Kutta methods can be more efficient when comparing their accuracy to the number of function calls (amount of acceleration solutions calculated), as they can use much larger step-sizes with a few more stages. In order to perform the measurement updates, our EKF requires regular satellite state solutions. Thus the only effect of implementing higher order methods will be an increase of stages to calculate with no justifiable improvements. In their real-time OD case studies, Montenbruck [13] used the 4^{th} order Runge-Kutta method for its step size flexibility and minimum memory requirements.

As we are dealing with a special case where the second half of our states (velocity) is the derivative of the first half of our states (position), the method is split up into two smaller problems that need to be solved

simultaneously (adapted from [55]). The RK4 method is summarised in Algorithm 5.2. A geometrical representation of the RK4 method (for a single state problem) can be found in Figure 5.2.

Algorithm 5.2 RK4: $\mathbf{r}_{k+1}, \dot{\mathbf{r}}_{k+1} \leftarrow \mathbf{r}_k, \dot{\mathbf{r}}_k, t_k, t_s$

$$\mathbf{k}_{1,r} = \dot{\mathbf{r}}_\mathbf{k} t_s$$
$$\mathbf{k}_{1,\dot{r}} = \mathbf{f}(\mathbf{r}_k, \dot{\mathbf{r}}_k, t_k) t_s$$
$$\mathbf{k}_{2,r} = \left(\dot{\mathbf{r}}_k + \frac{\mathbf{k}_{1,\dot{r}}}{2}\right) t_s$$
$$\mathbf{k}_{2,\dot{r}} = \mathbf{f}\left(\mathbf{r}_k + \frac{\mathbf{k}_{1,r}}{2}, \dot{\mathbf{r}}_k + \frac{\mathbf{k}_{1,\dot{r}}}{2}, t_k + \frac{t_s}{2}\right) t_s$$
$$\mathbf{k}_{3,r} = \left(\dot{\mathbf{r}}_k + \frac{\mathbf{k}_{2,\dot{r}}}{2}\right) t_s$$
$$\mathbf{k}_{3,\dot{r}} = \mathbf{f}\left(\mathbf{r}_k + \frac{\mathbf{k}_{2,r}}{2}, \dot{\mathbf{r}}_k + \frac{\mathbf{k}_{2,\dot{r}}}{2}, t_k + \frac{t_s}{2}\right) t_s$$
$$\mathbf{k}_{4,r} = \left(\dot{\mathbf{r}}_k + \mathbf{k}_{3,\dot{r}}\right) t_s$$
$$\mathbf{k}_{4,\dot{r}} = \mathbf{f}\left(\mathbf{r}_k + \mathbf{k}_{3,r}, \dot{\mathbf{r}}_k + \mathbf{k}_{3,\dot{r}}, t_k + t_s\right) t_s$$
$$\mathbf{r}_{k+1} = \tfrac{1}{6}\left(\mathbf{k}_{1,r} + 2\mathbf{k}_{2,r} + 2\mathbf{k}_{3,r} + \mathbf{k}_{4,r}\right)$$
$$\dot{\mathbf{r}}_{k+1} = \tfrac{1}{6}\left(\mathbf{k}_{1,\dot{r}} + 2\mathbf{k}_{2,\dot{r}} + 2\mathbf{k}_{3,\dot{r}} + \mathbf{k}_{4,\dot{r}}\right)$$

▷ where $\mathbf{f}(\mathbf{r}, \dot{\mathbf{r}})$ is the acceleration model, as described in Chapter 2.
▷ t_k is the timestamp at k, whereas t_s is the sample time ($t_s = t_{k+1} - t_k$).

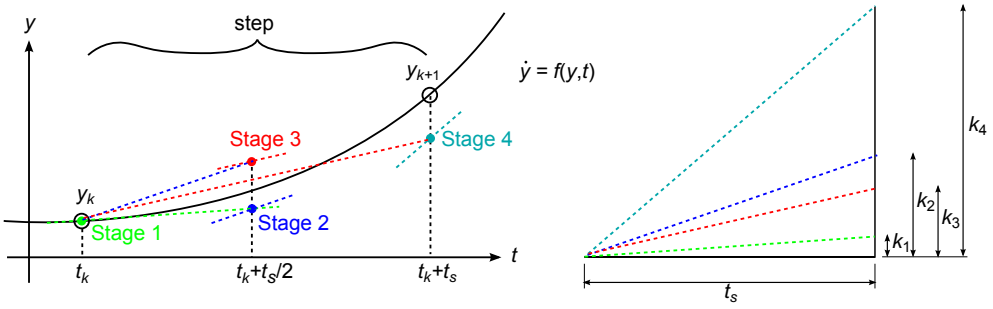


Figure 5.2 – Geometrical illustration of RK4 method.

5.3.2 Propagator Step-size

When it comes to numerical integration methods, the step-size plays a vital role. Three types of error in numerical methods are affected by step-size [56]:

- A *truncation* error originates from our numerical scheme (RK4 in this case) being a truncation of the Taylor series. This error decreases as step-size decreases.
- The fact that computers only have a finite word length is the cause for *round-off* errors. These are more prominent when small differences between large numbers are calculated, and consequently they increase as step-size decreases.
- *Inherited* error is the accumulation of errors over time - if the state is inaccurate, the model will use this defected state to calculate an incorrect state update.

A trade-off between truncation and round-off errors is thus required to yield reliable results. However, the extreme changes our states undergo for the typical step-size we use, along with the fact that we are using double precision floating point numbers, cause the round-off error to be less of a concern. Montenbruck

[13] used a 30 s step-size for their case studies, and we can assume this is a good starting point. This coincidentally happens to be the sampling rate of the SumbandilaSat GPS data.

The effect of step-size is tested through simulation. Note that this experiment used the full EKF, which is only explained later, to obtain initialising states. The EKF was provided with enough samples to settle, after which it propagated with its RK4 integrator (deprived of measurements). Step-sizes varying from 1 to 120 s were used. If we assume that the round-off error is negligible, we can expect the smallest step-size (i.e. 1 second in this case) to yield the most accurate solution. Thus the numerical method error is defined as the difference between the positions yielded by the propagator that used the step-size under inspection and the 1 second step-size propagator. Figure 5.3 presents the magnitude of the numerical method error, when using various step-sizes, for a 4-hour simulation. As expected, the error increases as step-size increases. It is clear that a step-size of 120 s will introduce unacceptable numerical errors. The 60 s step-size propagator reached an error of 30 m after one full orbit period, and it proceeded to grow exponentially after this. It seems that the 30 s step size will be the optimal choice, as it is accurate within 1 m for two full orbit periods. Any accuracy improvement beyond this level is negligible for our application.

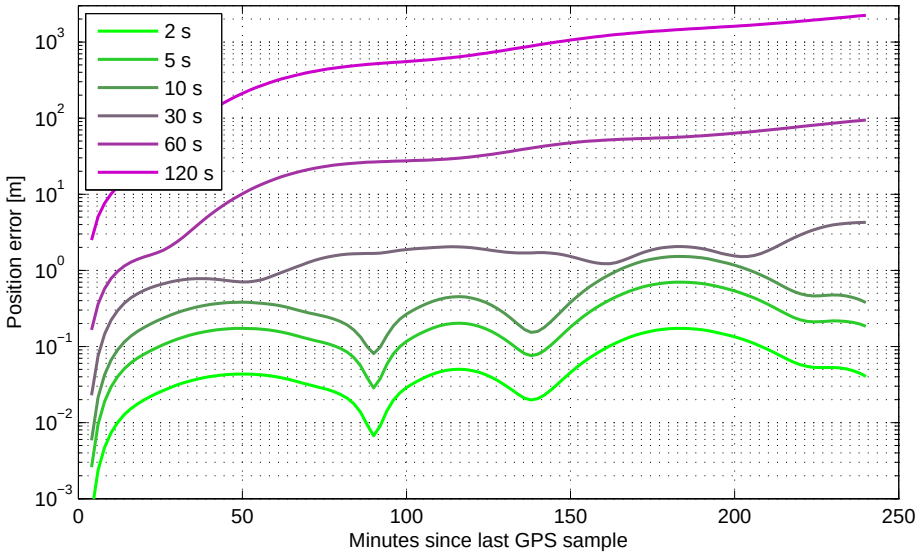


Figure 5.3 – RK4 truncation error when different step-sizes are used.

5.4 F and H Matrices

An EKF requires the determination of $\mathbf{F}$ and $\mathbf{H}$ matrices. These are obtained from the partial derivatives of the $\mathbf{f}(\mathbf{x}_t)$ and $\mathbf{h}(\mathbf{x}_t)$ vector equations. To implement an EKF on a microcontroller, a discrete versions of these matrices, namely $\mathbf{F}_k$ and $\mathbf{H}_k$, is required.

According to discretization theory, the $\mathbf{H}_k$ matrix is identical to its continuous version, $\mathbf{H}_t$. For this EKF, both have the trivial solution of the identity matrix, as our GPS receiver outputs the Cartesian position and velocity vectors (these just need to be transformed from the ECEF to the ECI frame), which are the states as defined in Equation 5.1. This presents the advantage of having no linearisation error when calculating the observation partial derivatives [48].

$$\mathbf{H}_k = \mathbf{H}_t = \frac{\partial \mathbf{h}(\mathbf{x}_t)}{\partial \mathbf{x}_t} = \mathbf{I} \tag{5.3}$$

The discrete $\mathbf{F}_k$ matrix is obtained through truncation of the Taylor expansion:

$$\mathbf{F}_k \approx \mathbf{I} + \mathbf{F}_t t_s \tag{5.4}$$

where t_s is the sample time. $\mathbf{F}_t$ is defined in Equation 5.5. The derivation of these entries is discussed in Subsections 5.4.1 to 5.4.4. If one wants to estimate SRP and drag coefficients (as done in some the studies discussed in Section 5.1), the $\mathbf{F}_t$ matrix would increase by one order for each extra state to be estimated.

$$\mathbf{F}_t = \frac{\partial \mathbf{f}(\mathbf{x}_t)}{\partial \mathbf{x}_t} = \begin{bmatrix} \frac{\partial \dot{x}}{\partial x} & \frac{\partial \dot{x}}{\partial y} & \frac{\partial \dot{x}}{\partial z} & \frac{\partial \dot{x}}{\partial \dot{x}} & \frac{\partial \dot{x}}{\partial \dot{y}} & \frac{\partial \dot{x}}{\partial \dot{z}} \\ \frac{\partial \dot{y}}{\partial x} & \frac{\partial \dot{y}}{\partial y} & \frac{\partial \dot{y}}{\partial z} & \frac{\partial \dot{y}}{\partial \dot{x}} & \frac{\partial \dot{y}}{\partial \dot{y}} & \frac{\partial \dot{y}}{\partial \dot{z}} \\ \frac{\partial \dot{z}}{\partial x} & \frac{\partial \dot{z}}{\partial y} & \frac{\partial \dot{z}}{\partial z} & \frac{\partial \dot{z}}{\partial \dot{x}} & \frac{\partial \dot{z}}{\partial \dot{y}} & \frac{\partial \dot{z}}{\partial \dot{z}} \\ \frac{\partial \ddot{x}}{\partial x} & \frac{\partial \ddot{x}}{\partial y} & \frac{\partial \ddot{x}}{\partial z} & \frac{\partial \ddot{x}}{\partial \dot{x}} & \frac{\partial \ddot{x}}{\partial \dot{y}} & \frac{\partial \ddot{x}}{\partial \dot{z}} \\ \frac{\partial \ddot{y}}{\partial x} & \frac{\partial \ddot{y}}{\partial y} & \frac{\partial \ddot{y}}{\partial z} & \frac{\partial \ddot{y}}{\partial \dot{x}} & \frac{\partial \ddot{y}}{\partial \dot{y}} & \frac{\partial \ddot{y}}{\partial \dot{z}} \\ \frac{\partial \ddot{z}}{\partial x} & \frac{\partial \ddot{z}}{\partial y} & \frac{\partial \ddot{z}}{\partial z} & \frac{\partial \ddot{z}}{\partial \dot{x}} & \frac{\partial \ddot{z}}{\partial \dot{y}} & \frac{\partial \ddot{z}}{\partial \dot{z}} \end{bmatrix} \tag{5.5}$$

5.4.1 $\mathbf{F}_t$ Matrix Entries by Inspection

By inspection, the following $\mathbf{F}$ matrix elements have apparent values:

$$\begin{aligned} &\frac{\partial \dot{x}}{\partial \dot{x}} = \frac{\partial \dot{y}}{\partial \dot{y}} = \frac{\partial \dot{z}}{\partial \dot{z}} = 1 \\ &\frac{\partial \dot{x}}{\partial x} = \frac{\partial \dot{x}}{\partial y} = \frac{\partial \dot{x}}{\partial z} = \frac{\partial \dot{x}}{\partial \dot{y}} = \frac{\partial \dot{x}}{\partial \dot{z}} = 0 \\ &\frac{\partial \dot{y}}{\partial x} = \frac{\partial \dot{y}}{\partial y} = \frac{\partial \dot{y}}{\partial z} = \frac{\partial \dot{y}}{\partial \dot{x}} = \frac{\partial \dot{y}}{\partial \dot{z}} = 0 \\ &\frac{\partial \dot{z}}{\partial x} = \frac{\partial \dot{z}}{\partial y} = \frac{\partial \dot{z}}{\partial z} = \frac{\partial \dot{z}}{\partial \dot{x}} = \frac{\partial \dot{z}}{\partial \dot{y}} = 0 \end{aligned} \tag{5.6}$$

These are due to our system dynamics modelled as accelerations (from forces) only. As the gravity model (2-Body problem with additional J_2 to J_4 amendments) is a conservative force model, it will only occupy the $\frac{\partial \ddot{r}}{\partial r}$ parts of the $\mathbf{F}$ matrix. The solar radiation pressure and atmospheric drag parts, however, will occupy both the $\frac{\partial \ddot{r}}{\partial r}$ and $\frac{\partial \ddot{r}}{\partial \dot{r}}$ parts of the $\mathbf{F}$ matrix.

5.4.2 2-Body Gravitational Model

From the 2-body problem acceleration in Equation 2.16:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix}_{2-body} = -\frac{\mu}{r^3}\mathbf{r} = \mu(x^2+y^2+z^2)^{-3/2} \begin{bmatrix} x \\ y \\ z \end{bmatrix} \tag{5.7}$$

The $\partial\ddot{x}$, $\partial\ddot{y}$ and $\partial\ddot{z}$ parts induced by the 2-body problem can now be determined:

$$\begin{aligned} &\frac{\partial \ddot{x}_{2-body}}{\partial x} = -\mu\left(\frac{-1}{r^3} + \frac{3x^2}{r^5}\right) \\ &\frac{\partial \ddot{x}_{2-body}}{\partial y} = -\mu\left(\frac{3xy}{r^5}\right) \\ &\frac{\partial \ddot{x}_{2-body}}{\partial z} = \frac{z}{y}\frac{\partial \ddot{x}_{2-body}}{\partial y} \end{aligned} \tag{5.8}$$

$$
\begin{aligned}
\frac{\partial \ddot{y}_{2-body}}{\partial x} &= \frac{\partial \ddot{x}_{2-body}}{\partial y} \\
\frac{\partial \ddot{y}_{2-body}}{\partial y} &= -\mu\left(\frac{-1}{r^3} + \frac{3y^2}{r^5}\right) \\
\frac{\partial \ddot{y}_{2-body}}{\partial z} &= \frac{z}{x}\frac{\partial \ddot{y}_{2-body}}{\partial x} \\
\frac{\partial \ddot{z}_{2-body}}{\partial x} &= \frac{\partial \ddot{x}_{2-body}}{\partial z} \\
\frac{\partial \ddot{z}_{2-body}}{\partial y} &= \frac{\partial \ddot{y}_{2-body}}{\partial z} \\
\frac{\partial \ddot{z}_{2-body}}{\partial z} &= -\mu\left(\frac{-1}{r^3} + \frac{3z^2}{r^5}\right)
\end{aligned}
\tag{5.8}
$$

5.4.3 J_2, J_3 and J_4 Amendments of Gravitational Model

Accelerations induced on a satellite by the J_2 to J_4 terms are described by Equations 2.17 to 2.19. By expanding these equations the following can be obtained:

$$
\begin{aligned}
\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}_{J_2} &= \frac{3J_2\mu R_e^2}{2}\begin{bmatrix} xz^2(x^2+y^2+z^2)^{-7/2} - x(x^2+y^2+z^2)^{-5/2} \\ yz^2(x^2+y^2+z^2)^{-7/2} - y(x^2+y^2+z^2)^{-5/2} \\ 5z^3(x^2+y^2+z^2)^{-7/2} - 3z(x^2+y^2+z^2)^{-5/2}\end{bmatrix} \\
\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}_{J_3} &= \frac{5J_3\mu R_e^3}{2}\begin{bmatrix} -3xz(x^2+y^2+z^2)^{-7/2} + 7xz^3(x^2+y^2+z^2)^{-9/2} \\ -3yz(x^2+y^2+z^2)^{-7/2} + 7yz^3(x^2+y^2+z^2)^{-9/2} \\ \frac{3}{5}(x^2+y^2+z^2)^{-5/2} - 6z^2(x^2+y^2+z^2)^{-7/2} + 7z^4(x^2+y^2+z^2)^{-9/2}\end{bmatrix} \\
\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix}_{J_4} &= \frac{15J_4\mu R_e^4}{8}\begin{bmatrix} x(x^2+y^2+z^2)^{-7/2} - 14xz^2(x^2+y^2+z^2)^{-9/2} + 21xz^4(x^2+y^2+z^2)^{-11/2} \\ y(x^2+y^2+z^2)^{-7/2} - 14yz^2(x^2+y^2+z^2)^{-9/2} + 21yz^4(x^2+y^2+z^2)^{-11/2} \\ 5z(x^2+y^2+z^2)^{-7/2} - \frac{70}{3}z^3(x^2+y^2+z^2)^{-9/2} + 21z^5(x^2+y^2+z^2)^{-11/2}\end{bmatrix}
\end{aligned}
\tag{5.9}
$$

We differentiate the J_2 accelerations in order to obtain the following corresponding **F** matrix entries.

$$
\begin{aligned}
\frac{\partial \ddot{x}_{J_2}}{\partial x} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{-1}{r^5} + \frac{5x^2+5z^2}{r^7} - \frac{35x^2z^2}{r^9}\right) \\
\frac{\partial \ddot{x}_{J_2}}{\partial y} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{-35z^2}{r^9} + \frac{5}{r^7}\right)xy \\
\frac{\partial \ddot{x}_{J_2}}{\partial z} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{15}{r^7} - \frac{35z^2}{r^9}\right)xz \\
\frac{\partial \ddot{y}_{J_2}}{\partial x} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{-35z^2}{r^9} + \frac{5}{r^7}\right)xy \\
\frac{\partial \ddot{y}_{J_2}}{\partial y} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{-1}{r^5} + \frac{5y^2+5z^2}{r^7} - \frac{35y^2z^2}{r^9}\right) \\
\frac{\partial \ddot{y}_{J_2}}{\partial z} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{15}{r^7} - \frac{35z^2}{r^9}\right)yz \\
\frac{\partial \ddot{z}_{J_2}}{\partial x} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{-15}{r^7} + \frac{35z^2}{r^9}\right)xz \\
\frac{\partial \ddot{z}_{J_2}}{\partial y} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{-15}{r^7} + \frac{35z^2}{r^9}\right)yz \\
\frac{\partial \ddot{z}_{J_2}}{\partial z} &= \frac{3J_2\mu R_e^2}{2}\left(\frac{-3}{r^5} + \frac{30z^2}{r^7} - \frac{35z^4}{r^9}\right)
\end{aligned}
\tag{5.10}
$$

Notice the similarities that can be exploited:

$$\begin{aligned}
\frac{\partial \ddot{y}_{J_2}}{\partial x} &= \frac{\partial \ddot{x}_{J_2}}{\partial y} & \frac{\partial \ddot{y}_{J_2}}{\partial z} &= \frac{y}{x}\frac{\partial \ddot{x}_{J_2}}{\partial z} \\
\frac{\partial \ddot{z}_{J_2}}{\partial x} &= \frac{\partial \ddot{x}_{J_2}}{\partial z} & \frac{\partial \ddot{z}_{J_2}}{\partial z} &= \frac{\partial \ddot{y}_{J_2}}{\partial z}
\end{aligned} \tag{5.11}$$

Following the same argument as with the J_2 acceleration parts of the $\mathbf{F}$ matrix, the following entries for the J_3 and J_4 terms can be obtained. Note how the similarities are exploited again.

$$\begin{aligned}
\frac{\partial \ddot{x}_{J_3}}{\partial x} &= \frac{5J_3\mu R_e^3}{2}\left(\frac{-3}{r^7} + \frac{35x^2+7z^2}{r^9} - \frac{63x^2z^2}{r^{11}}\right)z \\
\frac{\partial \ddot{x}_{J_3}}{\partial y} &= \frac{5J_3\mu R_e^3}{2}\left(\frac{35}{r^9} - \frac{63z^2}{r^{11}}\right)xyz \\
\frac{\partial \ddot{x}_{J_3}}{\partial z} &= \frac{5J_3\mu R_e^3}{2}\left(\frac{-3}{r^7} + \frac{42z^2}{r^9} - \frac{63z^4}{r^{11}}\right)x \\
\frac{\partial \ddot{y}_{J_3}}{\partial x} &= \frac{\partial \ddot{x}_{J_3}}{\partial y} \\
\frac{\partial \ddot{y}_{J_3}}{\partial y} &= \frac{5J_3\mu R_e^3}{2}\left(\frac{-3}{r^7} + \frac{35y^2+7z^2}{r^9} - \frac{63y^2z^2}{r^{11}}\right)z \\
\frac{\partial \ddot{y}_{J_3}}{\partial z} &= \frac{y}{x}\frac{\partial \ddot{x}_{J_3}}{\partial z} \\
\frac{\partial \ddot{z}_{J_3}}{\partial x} &= \frac{5J_3\mu R_e^3}{2}\left(\frac{-3}{r^7} + \frac{42z^2}{r^9} - \frac{63z^4}{r^{11}}\right)x = \frac{\partial \ddot{x}_{J_3}}{\partial z} \\
\frac{\partial \ddot{z}_{J_3}}{\partial y} &= \frac{y}{x}\frac{\partial \ddot{z}_{J_3}}{\partial x} \\
\frac{\partial \ddot{z}_{J_3}}{\partial z} &= \frac{5J_3\mu R_e^3}{2}\left(\frac{-15}{r^7} + \frac{70z^2}{r^9} - \frac{63z^4}{r^{11}}\right)z
\end{aligned} \tag{5.12}$$

$$\begin{aligned}
\frac{\partial \ddot{x}_{J_4}}{\partial x} &= \frac{15J_4\mu R_e^4}{8}\left(\frac{1}{r^7} - \frac{7x^2+14z^2}{r^9} + \frac{126x^2z^2+21z^4}{r^{11}} - \frac{231x^2z^4}{r^{13}}\right)z \\
\frac{\partial \ddot{x}_{J_4}}{\partial y} &= \frac{15J_4\mu R_e^4}{8}\left(\frac{-7}{r^9} + \frac{126z^2}{r^{11}} - \frac{231z^4}{r^{13}}\right)xy \\
\frac{\partial \ddot{x}_{J_4}}{\partial z} &= \frac{15J_4\mu R_e^4}{8}\left(\frac{-21}{r^9} + \frac{210z^2}{r^{11}} - \frac{231z^4}{r^{13}}\right)xz \\
\frac{\partial \ddot{y}_{J_4}}{\partial x} &= \frac{\partial \ddot{x}_{J_4}}{\partial y} \\
\frac{\partial \ddot{y}_{J_4}}{\partial y} &= \frac{15J_4\mu R_e^4}{8}\left(\frac{1}{r^7} - \frac{7y^2+14z^2}{r^9} + \frac{126y^2z^2+21z^4}{r^{11}} - \frac{231y^2z^4}{r^{13}}\right)z \\
\frac{\partial \ddot{y}_{J_4}}{\partial z} &= \frac{y}{x}\frac{\partial \ddot{x}_{J_4}}{\partial z} \\
\frac{\partial \ddot{z}_{J_4}}{\partial x} &= \frac{15J_4\mu R_e^4}{8}\left(\frac{-35}{r^9} + \frac{210z^2}{r^{11}} - \frac{231z^4}{r^{13}}\right)xz \\
\frac{\partial \ddot{z}_{J_4}}{\partial y} &= \frac{y}{x}\frac{\partial \ddot{z}_{J_4}}{\partial x} \\
\frac{\partial \ddot{z}_{J_4}}{\partial z} &= \frac{15J_4\mu R_e^4}{8}\left(\frac{5}{r^7} - \frac{105z^2}{r^9} + \frac{315z^4}{r^{11}} - \frac{231z^6}{r^{13}}\right)
\end{aligned} \tag{5.13}$$

Cunningham [18] shows how $\mathbf{F}$ matrix entries for the gravity perturbations can be obtained by an iterative process similar to that of Algorithm 2.2. It will later be shown that such a detailed $\mathbf{F}$ matrix is not required for our application.

5.4.4 Atmospheric Drag

Equation 2.45 described how we obtain the Earth-fixed velocity used for aerodynamic drag acceleration. However, our state of the satellite is defined in terms of the ECI frame, and this complicates the **F** matrix entries for drag. Expanding Equation 2.38 gives:

$$\begin{aligned}
\ddot{x}_{drag} &= \alpha_{drag}|\mathbf{v}_{aero}|\dot{x}_{aero} \\
&= \alpha_{drag}\sqrt{\dot{x}_{aero}^2 + \dot{y}_{aero}^2 + \dot{z}_{aero}^2}\dot{x}_{aero} \\
&= \alpha_{drag}\sqrt{(\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2}(\dot{x}+\omega_e y) \\
\ddot{y}_{drag} &= \alpha_{drag}\sqrt{(\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2}(\dot{y}-\omega_e x) \\
\ddot{z}_{drag} &= \alpha_{drag}\sqrt{(\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2}(\dot{z})
\end{aligned} \tag{5.14}$$

We can see clearly that the drag accelerations depend on all the states except for position z. We can now determine the following **F** matrix entries:

$$\begin{aligned}
\frac{\partial \ddot{x}_{drag}}{\partial x} &= \alpha_{drag}(-\omega_e)(2)(\dot{y}-\omega_e x)\frac{1}{2}\left((\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2\right)^{-\frac{1}{2}}(\dot{x}+\omega_e y) \\
&= -\alpha_{drag}\omega_e \frac{\dot{x}_{aero}\dot{y}_{aero}}{|\mathbf{v}_{aero}|} \\
\frac{\partial \ddot{x}_{drag}}{\partial y} &= \alpha_{drag}\Bigg(\omega_e(2)(\dot{x}+\omega_e y)\frac{1}{2}\left((\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2\right)^{-\frac{1}{2}}(\dot{x}+\omega_e y) \\
&\qquad + \left((\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2\right)^{\frac{1}{2}}(\dot{x}+\omega_e y)\Bigg) \\
&= \alpha_{drag}\omega_e\left(\frac{\dot{x}_{aero}^2}{|\mathbf{v}_{aero}|} + |\mathbf{v}_{aero}|\right) \\
\frac{\partial \ddot{x}_{drag}}{\partial z} &= 0 \\
\frac{\partial \ddot{x}_{drag}}{\partial \dot{x}} &= \alpha_{drag}\Bigg(\sqrt{(\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2} \\
&\qquad + (2)\frac{1}{2}(\dot{x}+\omega_e y)(\dot{x}+\omega_e y)\left((\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2\right)^{-\frac{1}{2}}\Bigg) \\
&= \alpha_{drag}\left(|\mathbf{v}_{aero}| + \frac{\dot{x}_{aero}^2}{|\mathbf{v}_{aero}|}\right) \\
&= \frac{1}{\omega_e}\frac{\partial \ddot{x}_{drag}}{\partial y} \\
\frac{\partial \ddot{x}_{drag}}{\partial \dot{y}} &= \alpha_{drag}(2)\frac{1}{2}(\dot{y}-\omega_e x)(\dot{x}+\omega_e y)\left((\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2\right)^{-\frac{1}{2}} \\
&= \alpha_{drag}\frac{\dot{x}_{aero}\dot{y}_{aero}}{|\mathbf{v}_{aero}|} \\
&= \frac{-1}{\omega_e}\frac{\partial \ddot{x}_{drag}}{\partial x} \\
\frac{\partial \ddot{x}_{drag}}{\partial \dot{z}} &= \alpha_{drag}(2)\frac{1}{2}\dot{z}(\dot{x}+\omega_e y)\left((\dot{x}+\omega_e y)^2 + (\dot{y}-\omega_e x)^2 + \dot{z}^2\right)^{-\frac{1}{2}} \\
&= \alpha_{drag}\frac{\dot{x}_{aero}\dot{z}}{|\mathbf{v}_{aero}|}
\end{aligned} \tag{5.15}$$

Following the same steps as above and realising some similarities, the rest of the **F** matrix entries are:

$$
\begin{aligned}
\frac{\partial \ddot{y}_{drag}}{\partial x} &= -\alpha_{drag}\omega_e\left(\frac{\dot{y}_{aero}^2}{|\mathbf{v}_{aero}|}+|\mathbf{v}_{aero}|\right) & \frac{\partial \ddot{z}_{drag}}{\partial x} &= -\alpha_{drag}\omega_e\frac{\dot{y}_{aero}\dot{z}}{|\mathbf{v}_{aero}|}\\
\frac{\partial \ddot{y}_{drag}}{\partial y} &= \alpha_{drag}\omega_e\frac{\dot{x}_{aero}\dot{y}_{aero}}{|\mathbf{v}_{aero}|} & \frac{\partial \ddot{z}_{drag}}{\partial y} &= \alpha_{drag}\omega_e\frac{\dot{x}_{aero}\dot{z}}{|\mathbf{v}_{aero}|}\\
\frac{\partial \ddot{y}_{drag}}{\partial z} &= 0 & \frac{\partial \ddot{z}_{drag}}{\partial z} &= 0\\
\frac{\partial \ddot{y}_{drag}}{\partial \dot{x}} &= \alpha_{drag}\frac{\dot{x}_{aero}\dot{y}_{aero}}{|\mathbf{v}_{aero}|} & \frac{\partial \ddot{z}_{drag}}{\partial \dot{x}} &= \alpha_{drag}\frac{\dot{x}_{aero}\dot{z}}{|\mathbf{v}_{aero}|}\\
\frac{\partial \ddot{y}_{drag}}{\partial \dot{y}} &= \alpha_{drag}\left(|\mathbf{v}_{aero}|+\frac{\dot{y}_{aero}^2}{|\mathbf{v}_{aero}|}\right) & \frac{\partial \ddot{z}_{drag}}{\partial \dot{y}} &= \alpha_{drag}\frac{\dot{y}_{aero}\dot{z}}{|\mathbf{v}_{aero}|}\\
\frac{\partial \ddot{y}_{drag}}{\partial \dot{z}} &= \alpha_{drag}\frac{\dot{y}_{aero}\dot{z}}{|\mathbf{v}_{aero}|} & \frac{\partial \ddot{z}_{drag}}{\partial \dot{z}} &= \alpha_{drag}\left(\frac{\dot{z}^2}{|\mathbf{v}_{aero}|}+|\mathbf{v}_{aero}|\right)
\end{aligned}
\tag{5.16}
$$

For faster computation, the following relations can be used:

$$
\begin{aligned}
\frac{\partial \ddot{x}_{drag}}{\partial \dot{x}} &= \frac{1}{\omega_e}\frac{\partial \ddot{x}_{drag}}{\partial y} & \frac{\partial \ddot{x}_{drag}}{\partial \dot{y}} &= \frac{-1}{\omega_e}\frac{\partial \ddot{x}_{drag}}{\partial x} & \frac{\partial \ddot{y}_{drag}}{\partial y} &= -\frac{\partial \ddot{x}_{drag}}{\partial x}\\
\frac{\partial \ddot{y}_{drag}}{\partial \dot{x}} &= \frac{\partial \ddot{x}_{drag}}{\partial \dot{y}} & \frac{\partial \ddot{y}_{drag}}{\partial \dot{y}} &= \frac{-1}{\omega_e}\frac{\partial \ddot{y}_{drag}}{\partial x} & \frac{\partial \ddot{z}_{drag}}{\partial x} &= -\omega_e\frac{\partial \ddot{y}_{drag}}{\partial \dot{z}}\\
\frac{\partial \ddot{z}_{drag}}{\partial y} &= \omega_e\frac{\partial \ddot{x}_{drag}}{\partial \dot{z}} & \frac{\partial \ddot{z}_{drag}}{\partial \dot{x}} &= \frac{\partial \ddot{x}_{drag}}{\partial \dot{z}} & \frac{\partial \ddot{z}_{drag}}{\partial \dot{y}} &= \frac{\partial \ddot{y}_{drag}}{\partial \dot{y}}
\end{aligned}
\tag{5.17}
$$

5.5 System and Measurement Noises

Uncertainties of the orbit dynamics originate from incomplete or inaccurate force models. Forces are directly linked to accelerations, and therefore system noise will be modelled as disturbances on accelerations only (the derivative of the velocity state).

The most considerable obstacle in the implementation of a LEO GPS receiver is the extreme dynamics encountered: LEO satellites travel at 7 km/s in inertial space. Not only does this complicate the GPS satellite lock-on acquisition process, but it also makes in-track position measurement very vulnerable to inaccurate timestamps and other noises. A 1 second error in timestamp can result to a 7 km in-track error. We expect the largest error in measurement to originate from this phenomenon. See Appendix B for a more detailed discussion.

Let $\mathbf{Q}$ and $\mathbf{R}$ respectively represent the discrete system and measurement noise covariance matrices. According to Franklin and Powell [57] it is safe to assume that the system and measurement noises are uncorrelated between the different states, and therefore $\mathbf{Q}$ and $\mathbf{R}$ can be considered as diagonal matrices. Montenbruck [3] and Choi [48] made similar assumptions. This assumption will dramatically decrease computation and memory requirements, as the $\mathbf{Q}$ and $\mathbf{R}$ information can be stored and operated by means of vectors, instead of full 6×6 matrices. Define $\mathbf{Q}$ and $\mathbf{R}$ as:

$$
\begin{aligned}
\mathbf{Q} &= \mathrm{diag}(0,0,0,q,q,q)\\
\mathbf{R} &= \mathrm{diag}(r_r,r_r,r_r,r_v,r_v,r_v)
\end{aligned}
\tag{5.18}
$$

where q, r_r and r_v represent the expected noise variances of the acceleration model and the position and velocity measurement error variances. The zeros in $\mathbf{Q}$ are due to the model noise modelled as disturbance forces only. q is usually chosen to match the expected variance of the disturbance forces. But this is based on the assumption that measurements will be readily available. As this is not true in our case, we must ensure that our EKF is fast enough to converge to the true states in a measurement period.

We therefore select an arbitrary $q = 3.3 \times 10^{-4}$ as this value demonstrated the desired behaviour. The expected measurement noise covariances are discussed in Appendix B.8. Subsequently, we let:

$$\begin{aligned} r_r &= \sigma_r^2 = 101.2 \text{ m}^2 \\ r_v &= \sigma_v^2 = 0.2708 \text{ m}^2/\text{s}^2 \\ q &= 3.3 \times 10^{-4} \text{ m}^2/\text{s}^4 \end{aligned} \tag{5.19}$$

5.6 EKF Simulation Using SumbandilaSat GPS Data

The EKF estimator was implemented using the provided SumbandilaSat GPS data. A Matlab script was used to read the GPS data and feed it to the EKF, which was written in C and compiled to fit into the Matlab environment. All the graph plots were done in Matlab.

Propagation was done using a RK4 integrator, as described in Section 5.3. A JGM-3 gravity model with an adjustable order (maximum 40), along with basic atmospheric drag and SRP models, were used for the acceleration model. The satellite orbit EKF, as described in Algorithm 5.1, requires the inversion of $\mathbf{L}_k$, which is a 6×6 matrix. Matlab has built in functions to perform this with ease, but the EKF needs to be implemented in C. The classic analytical solution using Cramer's rule will cripple most systems. Instead, a blockwise inversion scheme, described in Appendix C.2, is used.

We know that the SumbandilaSat dataset contains some corrupt measurements. To avoid these, the EKF was set up to calculate the 3D difference between its propagated state and the GPS sample just before the measurement update. If this error is greater than a certain threshold, then this sample will be ignored (the measurement update is skipped). A suitable threshold for the SumbandilaSat dataset is 3000 m. Note that even though the EKF avoided these corrupt samples, the plotting script did not, and thus these are still present in the error plots.

Figure 5.4 shows the EKF performance for the first 60 minutes. A gravity model of order 40 was used. We can clearly see that the EKF is functioning properly. The error throughout the entire simulation is plotted in Figure 5.5. Settling is observed for the initialisation (during the first 5 minutes).

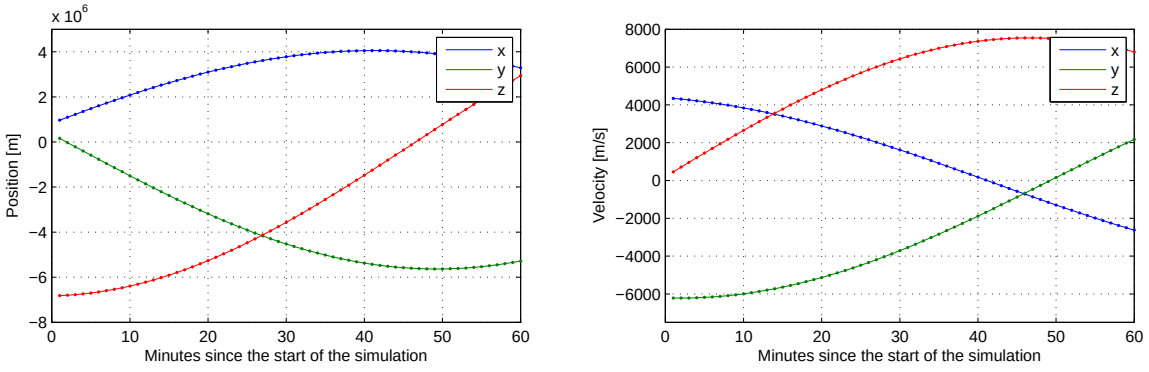


Figure 5.4 – EKF position (left) and velocity (right) states (lines) compared to GPS measurements (dots) during simulation of SumbandilaSat.

The goal for this thesis is to develop a system that can propagate the satellite's position with aid of the GPS for only a small portion of the orbit. The EKF algorithm is proficient at estimation, even if the system model is not extremely accurate. Our system, however, will not always be provided measurements, and thus we do need an accurate system model.

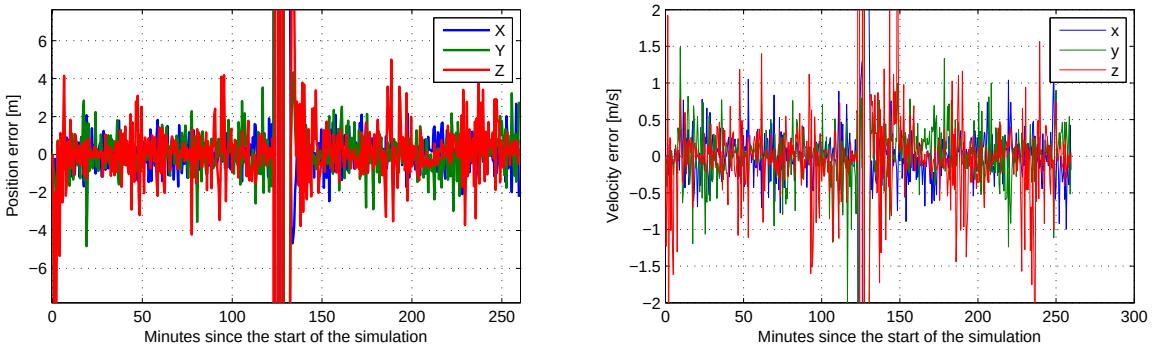


Figure 5.5 – Error of position (left) and velocity (right).

5.6.1 Propagation without Measurements

The propagator needs to function for most of the orbit without measurements. To test this scenario, the EKF is provided with the first 100 GPS measurements (which spans 50 minutes), from which it will propagate from the propagation update only. Figure 5.6 shows the position error. As expected, the in-track error dominates. A slight offset in the prediction of normal position results to a growth of in-track error. This is likely to happen as a very simplified atmospheric drag model is implemented and the gravity model is truncated. After a full orbit (about 95 minutes for SumbandilaSat) without measurements, the numerical propagator still sustained an accuracy of less than 150 m, and 250 m for two orbits.

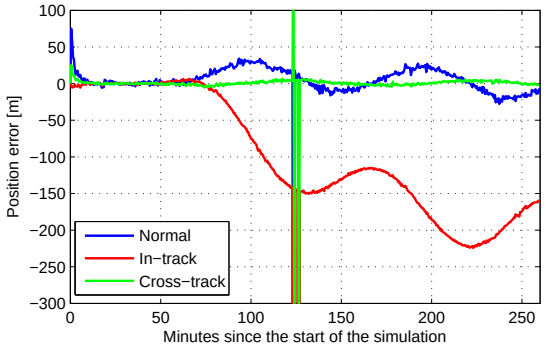


Figure 5.6 – NTW position error when the EKF is provided GPS measurements for 50 minutes.

5.6.2 Reactivating the GPS

The next step is to test the transient response of the EKF when we turn the GPS receiver back on. This test is done by providing the EKF with 10 minutes of GPS samples (20 measurements in SumbandilaSat's case) in every orbit period (of roughly 95 minutes). Figure 5.7 shows the position and velocity errors during simulation.

The simulation is repeated using the same dataset, but starting at different times in the dataset. This revealed that our system's performance varied (see Figure 5.8), which could indicate that the EKF is relying too much on the measurements and not filtering them enough.

Wanting to improve both the transient response and filtering capabilities of an EKF is like having your bread buttered on both sides. Simply changing the noise covariance matrices will not solve the problem, and thus we will need to look at some other strategies as well.

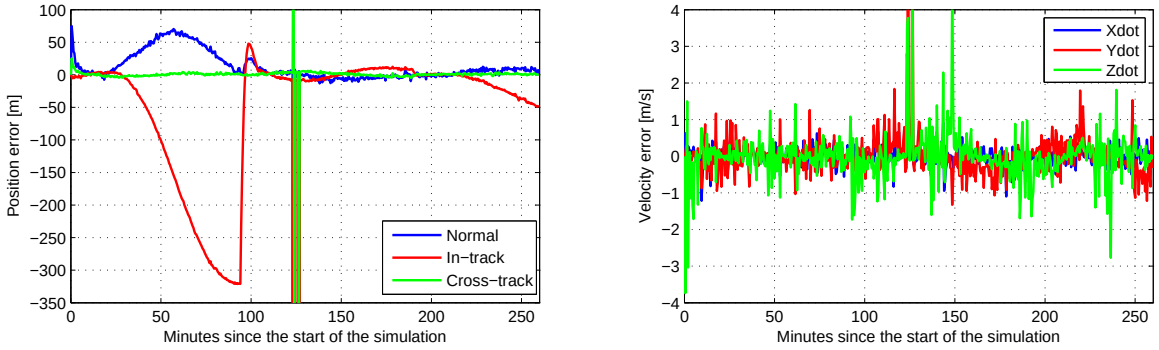


Figure 5.7 – NTW position and ECI velocity errors of EKF when provided with only 10 minutes of GPS samples every orbit.

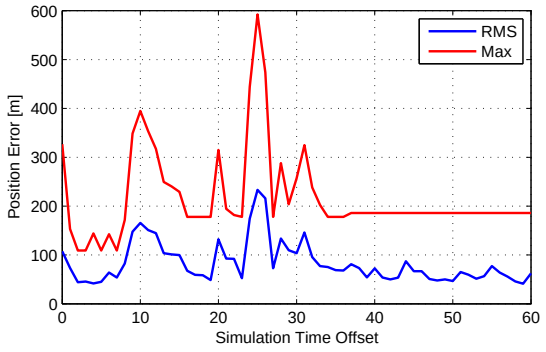


Figure 5.8 – RMS and maximum 3D position error for simulations with various starting time offsets.

5.7 Improving the EKF

The EKF's filtering abilities needs to be enhanced, but a satisfying transient response must still be maintained. The following strategies are proposed and investigated:

- Adjust and optimise the covariance matrices representing the expected model ($\mathbf{Q}$) and measurement ($\mathbf{R}$) noise. Traditionally, this is the starting point when attempting to improve a Kalman Filter. This will, however, still only be a trade-off between the transient response and filtering capabilities.
- Keep the GPS activated for a longer period of time (i.e. increase t_{act}). In order to keep within the power budget, t_{int} will also have to be increased. Optimal t_{act} and t_{int} values are investigated in Section 5.8.
- Increase the sampling rate of the GPS receiver.
- Force the EKF states to the GPS measurements every time the GPS receiver is reactivated.

5.7.1 Adjusting the EKF Noise Covariance Matrices

A fair approximation of the model and measurement noise magnitudes is essential for the success of a Kalman Filter. The noise magnitude approximations define the $\mathbf{Q}$ and $\mathbf{R}$ covariance matrices, which in turn are used to weigh our trust in the measurements to that of the model. If the $\mathbf{Q}$ to $\mathbf{R}$ ratio is too low, the EKF expects the model to be very close to perfect, and the EKF will ignore measurements. On the other hand, if this ratio is too high, the EKF will simply just follow the measurements and not filter

out the noise. This would lead to an arbitrary propagator performance varying to different GPS turn-off times, as the EKF will solely rely on the last measurement's accuracy.

Q Matrix

Increasing q might speed up the response of the EKF, but it will also deteriorate its filtering capabilities. To illustrate this behaviour, we will perform some simulations with faulty $\mathbf{Q}$ matrices. After propagating from the first 5 minutes of GPS measurements, the GPS is turned off for 45 minutes. Figure 5.9 shows the performance of the propagator when the GPS module is switched on again for various durations (as indicated by the legend). We first used $q = 1 \times 10^{-2}$ (left side of Figure 5.9). As the states rely almost entirely on the last measurement's accuracy, the performance of the EKF varies arbitrary to the choice of the last GPS measurement. Then we changed q to 1×10^{-6} (right side of Figure 5.9). It is clear that the EKF is very slow and requires too much time to settle.

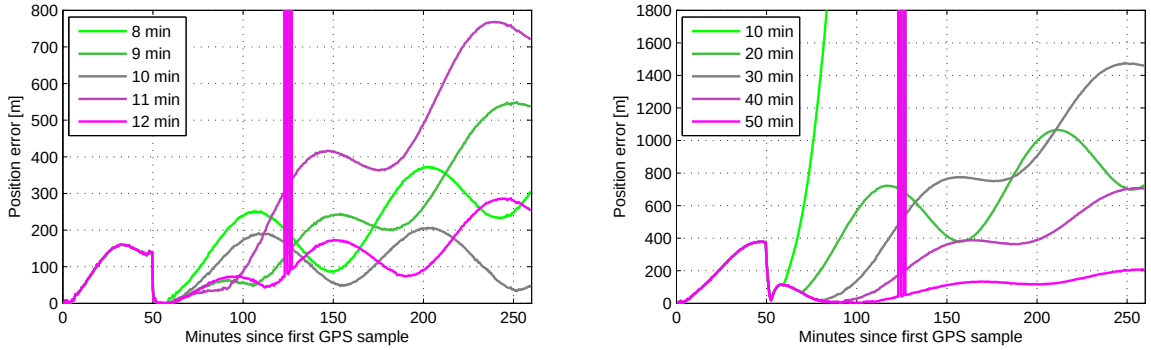


Figure 5.9 – Effect of incorrect $\mathbf{Q}$ matrix entries ($q = 1 \times 10^{-2}$ left and $q = 1 \times 10^{-6}$ right) on EKF performance.

R Matrix

To find the optimal measurement noise covariance matrix entries, we ran multiple simulations with varying r_r and r_v. To account for the variations due to simulation start time offsets, each r_r and r_v combination simulation was repeated using offset times ranging from 0 to 60 minutes in 5 minute intervals. The RMS and maximum position errors are plotted in Figure 5.10. It is clear that we need to use $r_r < 10$ for a safe region of operation. Our previous choice of $r_r \approx 100$ seems to have fallen into a narrow optimal region. When $r_r \leq 10$, choosing r_v does not make much of a difference. It must be noted that the measurement noise expectations were derived from Nortier's simulations. The simulations in this study are based on real-life hardware, which will most likely perform poorer, hence the difference between the expected and observed regions of r_r and r_v that exhibited the best accuracy. We will continue to work with the original r_r and r_v, as this test will be repeated after some other optimisation strategies have been put in place.

5.7.2 Increasing the Sampling Rate

It is intuitive that a faster sampling rate will yield a quicker transient response. As the tracking part of the active GPS receiver is predominantly responsible for the power consumption, increasing its sample rate of measurement extraction will have minimal effect on the power consumption. The faster sampling rate is only required for the measurement period, and thus we can continue with our 30 s sample rate when the GPS receiver is deactivated.

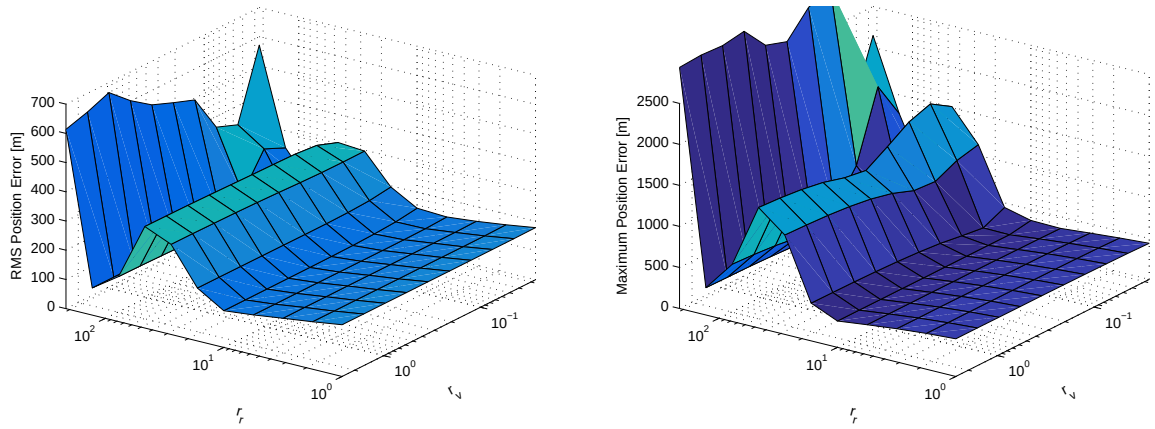


Figure 5.10 – EKF RMS and maximum position errors when r_r and r_v are varied.

Increasing a GPS receiver's sampling rate to 1 Hz is a straightforward task. However, as the GPS data we have obtained for this study has sample times of 30 s (SumbandilaSat) and 10 s (NigeriaSat), we will need to generate an artificial dataset to study the effect of an increased sample rate. We will superimpose Gaussian noise on the generated data to mimic realistic data, but note that the ultimate accuracy results from this subsection cannot be guaranteed.

Generating a Synthetic Dataset with a Higher Sample Rate

We can use the EKF to generate a dataset with a higher sampling rate, based on the original real data (the original data's sample time will from here on be referred to as the overarching sample time). This is done by performing the propagation update multiple times between measurement updates, saving the states after each propagation update. The real GPS measurements will be readily available, and thus we can reduce the EKF bandwidth for this application as the transient response is not a concern. This will increase the EKF's filter effect and thereby minimise the effects of the overarching sample rate. In other words, the effect of SumbandilaSat's new dataset jumping every 30 seconds to the real value will be mitigated. We must, however, not choose the $\mathbf{Q}$ arbitrarily too small, as the EKF might become too slow to follow the (real) measurements. From Subsection 5.3.2 we do not expect that decreasing the step size would have a negative effect on accuracy.

Using the method described above a 2 s sample time dataset was generated. We used $q = 1 \times 10^{-6}$ for the generating EKF. The position and velocity error during the upsampling process is shown in Figure 5.11. The EKF took 20 minutes to settle, and thus this first part of the data will not be used. Note that the (upsampling) EKF process will filter out the high frequency noises on the states, and appropriate noise must be added to the generated data. It is reasonable to assume that the errors in Figure 5.11 represent a good approximation of the GPS measurement noise we have filtered out. Table 5.1 summarises the properties of the errors from Figure 5.11. The in-track direction is defined to be aligned with the velocity

Table 5.1 – Properties of upsampling process errors.

	σ_{Normal}	$\sigma_{In-track}$	$\sigma_{Cross-track}$
Position (m)	2.424	0.981	0.953
Velocity (m/s)	0.611	0.229	0.239

vector of the propagator, and thus it might seem intuitive that the velocity errors will only be in the in-track

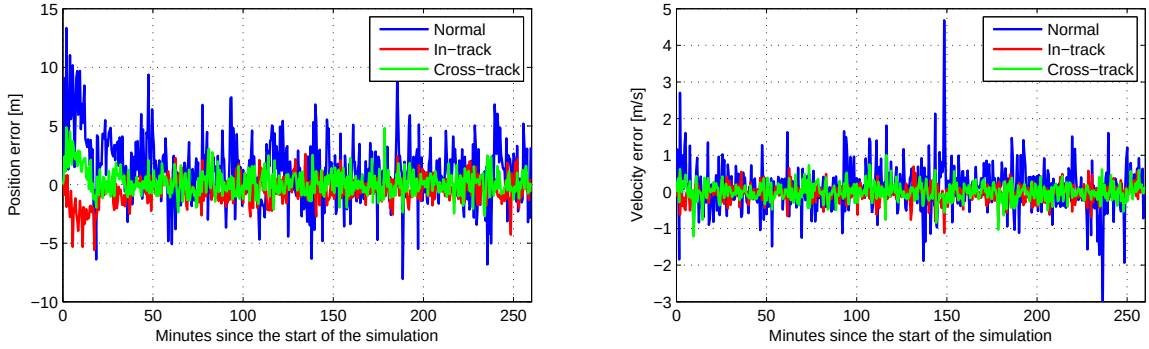


Figure 5.11 – Position and velocity errors of EKF during upsampling process.

direction. GPS measurements, however, can have noise in all directions. The noise to be superimposed is generated using Equation 5.20.

$$\mathbf{r}_{noise} = \begin{bmatrix} \mathbf{N} \\ \mathbf{T} \\ \mathbf{W} \end{bmatrix}^T \begin{bmatrix} N(0, 5.875) \\ N(0, 0.963) \\ N(0, 0.908) \end{bmatrix} \qquad \mathbf{v}_{noise} = \begin{bmatrix} \mathbf{N} \\ \mathbf{T} \\ \mathbf{W} \end{bmatrix}^T \begin{bmatrix} N(0, 0.373) \\ N(0, 0.053) \\ N(0, 0.057) \end{bmatrix} \tag{5.20}$$

where $N(0, \sigma^2)$ is a normally distributed random variable with zero mean and standard deviation σ.

Effect of Higher Sampling Rate

If we change the sampling rate, we need to recalculate the $\mathbf{Q}$ matrix entries. Assuming that $q \propto t_s$, we get $q = 3.3 \times 10^{-4} \times \frac{2}{30} = 2.2 \times 10^{-5}$.

The experiment of Figure 5.7 is repeated, and a typical simulation error plot is shown in Figure 5.12. It is clear that the transient response sped up, which can allow us to lower t_{act} and, subsequently, improve the power budget. The worst maximum 3D error was also reduced to 500 m. The RMS error, however, seems to have worsened slightly. From these simulations we can clearly see that a faster sampling rate can enhance the system's transient response. It must be noted that this Section's work does not guarantee a more accurate solution, as the dataset is synthetically generated.

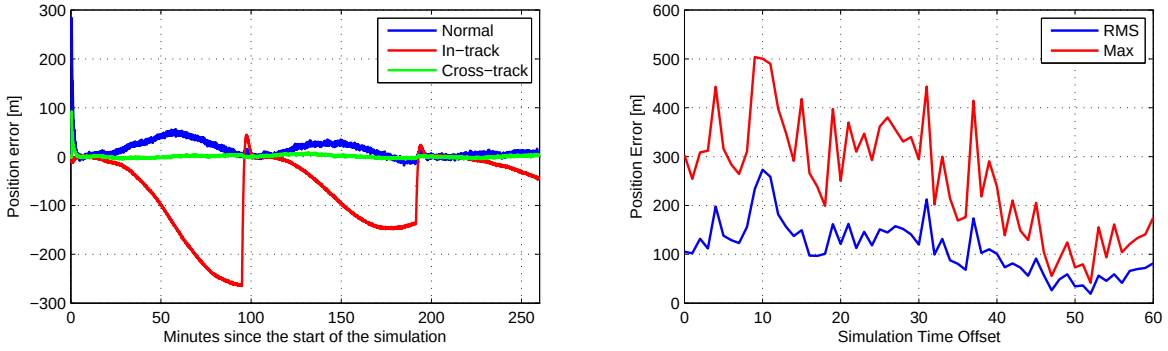


Figure 5.12 – Typical position error during higher sampling rate simulation (left), and a plot of maximum and RMS 3D position errors when different simulation starting time offsets were used (right).

5.7.3 Forcing the EKF States Upon GPS Activation

In the previous subsection we first improved the transient response by sampling faster, and then lowered the EKF's bandwidth for better noise cancellation, resulting in more accurate state estimation. We will now attempt the same, but this time we will simply force the EKF's states to the first measurement upon the GPS receiver's reactivation. As the transient response of the EKF is not of much concern now, we can lower the bandwidth by setting $q = 3.3 \times 10^{-5}$. The simulation of Figure 5.7 was repeated (using the original 30 s sample time data), with the new setup. The new position error is plotted in Figure 5.13. An astonishing improvement is clear - the 3D position error never exceeded 300 m. As this strategy proved to yield the best results, we will use this system in the work that is to come.

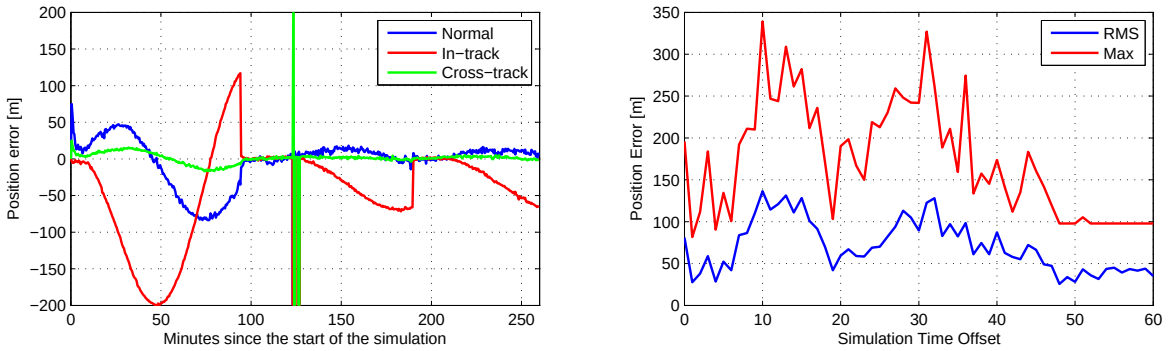


Figure 5.13 – Typical position error during a force-state system simulation (left), and a plot of maximum and RMS 3D position errors when simulation starting time was varied (right).

5.8 Optimising the EKF

Up until now our propagation has made use of a 40th order version of the JGM-3 Earth gravity model. The **F** matrix contained entries for J_2, J_3 and J_4 terms of the gravity model, as well as for aerodynamic drag. We will now eliminate some of these and see to what extent we can simplify our dynamics model. Remember that the system improvement described in Subsection 5.7.3 is used.

5.8.1 F Matrix

As the **F** matrix is not used to update the states, its complexity can be reduced. By using only the 2-body problem entries of Equation 5.8, the system's performance did not degrade. Figure 5.14 shows the performance of this reduced system, and it is clear that this simplification had almost no effect. Thus, from here on we will work with this simplified **F** matrix.

5.8.2 Complexity of Force Model

3rd Body Perturbations

As we are in the LEO environment, the Sun and the Moon gravitational perturbations will be small in comparison to aerodynamic drag and Earth gravity perturbations. The fact that we used a Taylor series approximation for these 3rd body perturbations (Section 2.6) also makes them less accurate. To test the importance of these perturbations, we ran simulations excluding each of the Sun and Moon's gravitational effects. The position errors are plotted in Figure 5.15. We see that excluding the Sun had almost no effect, whilst removing the Moon even resulted in better performance. Extending the Taylor series expansion

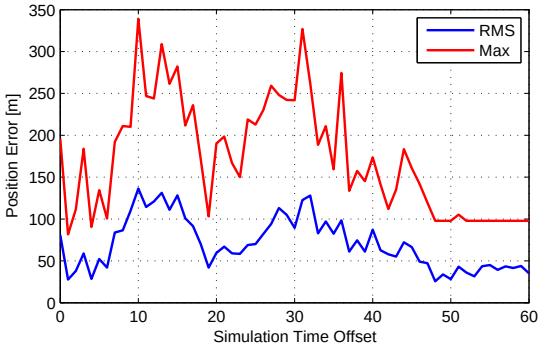


Figure 5.14 – Position error of EKF for different time offsets. Only 2-body equations used for **F** matrix.

could probably yield more accurate results, but as mentioned before, 3rd body perturbations are very small for LEO satellites, and the extra computations would not justify the almost negligible accuracy improvement. Both of these 3rd body perturbations are thus removed from our propagation model.

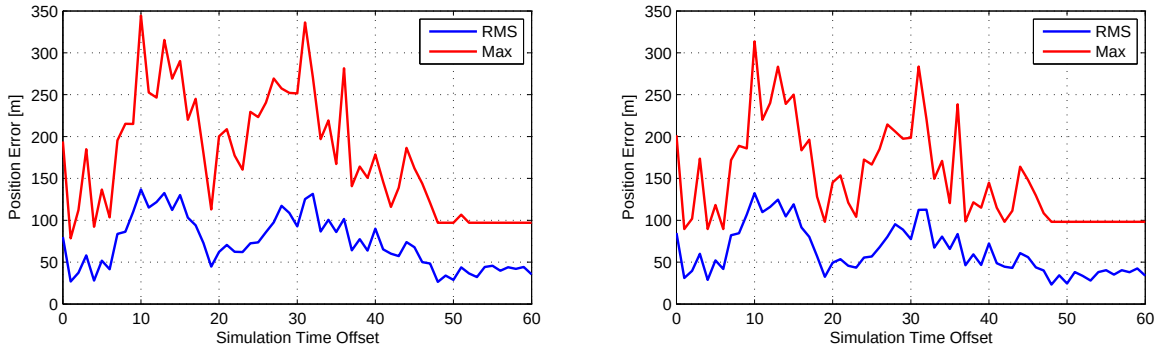


Figure 5.15 – Position error of EKF without the Sun (left) or Moon (right) gravitational pull perturbations.

SRP

Our SRP model is very basic. As aerodynamic drag and Earth gravity perturbations dominate the other disturbances, removing SRP from our propagator's force model had almost no effect on the system's performance. The position error is plotted in Figure 5.16. SRP is thus removed from our propagator from here on.

Aerodynamic Drag

Let's simplify our aerodynamic drag model to use a constant atmospheric density throughout the simulation. Figure 5.17 shows the EKF position error when we assume that the atmospheric density is constantly the mean of the SMAD exponential model (Figure 2.10) at 500 km altitude. Once again, the simplification's effect is negligible, and this change will be implemented from here on.

Gravity Model

The Earth gravity model is the most computational intensive part of our numerical integration propagation process, and thus minimising its order to an optimal point is crucial.

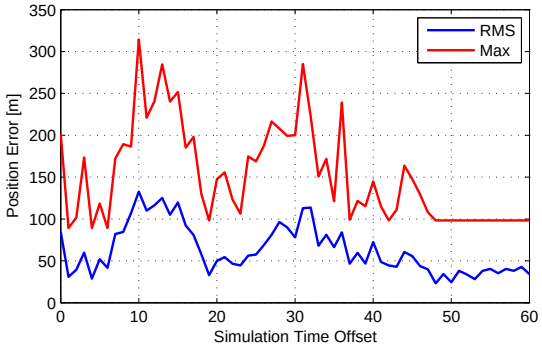


Figure 5.16 – Absolute position error of EKF without SRP.

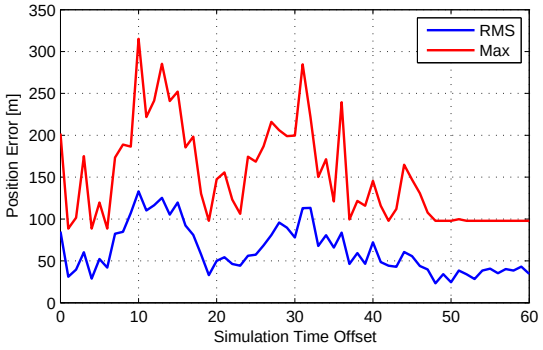


Figure 5.17 – Absolute position error of EKF with aerodynamic drag considered constant.

We simulated for different gravity model orders (and with time offsets that varied from 0 to 95 minutes in 2.5 minute intervals). Figure 5.18 shows the results. It is clear that for this SumbandilaSat simulation, a gravity order of more than 20 has very little effect on the propagator performance. This accuracy limit is due to an inevitably flawed aerodynamic drag model. According to this simulation, it seems that a 9^{th} order JGM-3 gravitational model would be sufficient. If a 24 hour GPS dataset was available, Figure 5.18 would probably have a closer resemblance to a decreasing exponential function.

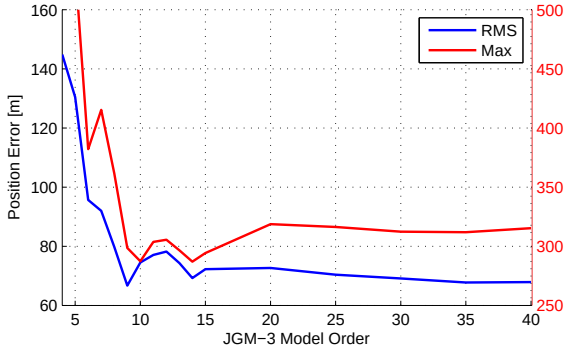


Figure 5.18 – Position error of EKF vs order of gravity model.

5.8.3 Optimal Covariance Matrices

Now that we have optimised the complexity of our EKF system, let's optimise for the best covariance matrix $\mathbf{R}$ (as described in Subsubsection 5.7.1). Figure 5.19 shows the 3D position errors when different r_r and r_v are implemented. We seem to obtain the best performance when $r_r \approx 50$ and $r_v \approx 0.8$, and thus we will use this configuration for what is to follow.

5.8.4 Regularity and Length of GPS Activations

To search for the optimal solution, we ran simulations varying both t_{act} and t_{int}. The errors are illustrated in Figure 5.20. Figure 5.21 shows contour plots of the same results. From the maximum error plot, an optimum seems to be when $t_{\text{act}} = 7$ min and $t_{\text{int}} = 75$ min. This configuration yielded RMS and maximum errors of approximately 60 m and 300 m, respectively.

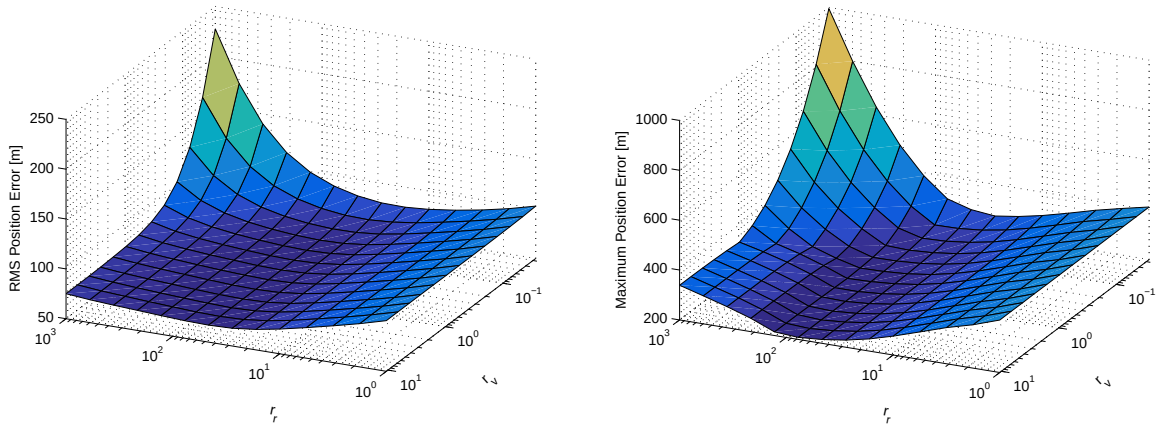


Figure 5.19 – Position error of EKF for various r_r and r_v.

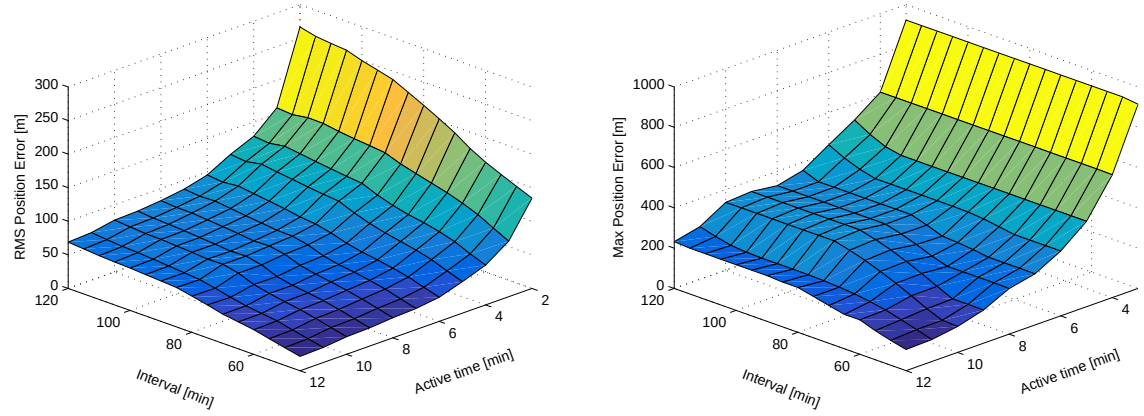


Figure 5.20 – Position error of EKF for various intervals and activation periods.

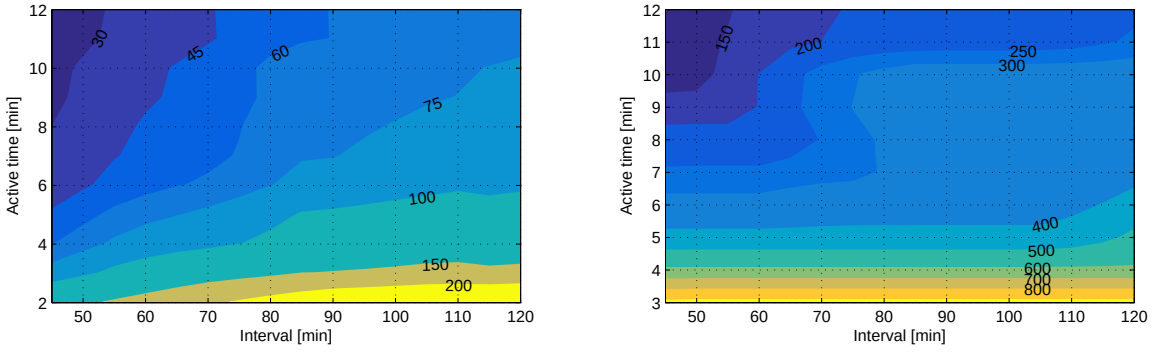


Figure 5.21 – Contour plot of Figure 5.20. RMS error is on left side and maximum error on right side.

5.9 Testing on Other GPS Datasets

We had to ensure that our optimisation was not just fitting the results to the specific SumbandilaSat data of 01/02/2010. Thus the system was tested with the other two provided GPS datasets (see Table 1.1). The test of Figure 5.13 is repeated, and position error of the EKF using the other SumbandilaSat dataset as input is plotted in Figure 5.22. The results are in accordance with the previous simulations done.

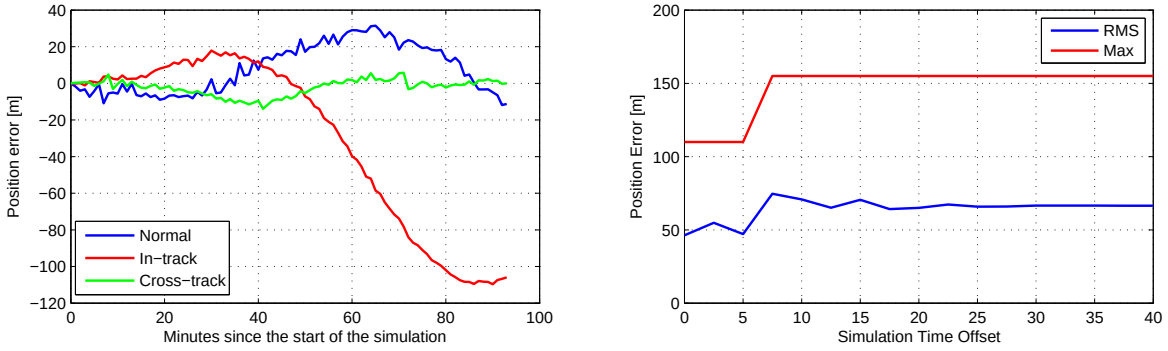


Figure 5.22 – EKF position error with SumbandilaSat GPS dataset of 07/02/2010 as input.

The NigeriaSat GPS data had a sample time of $t_s = 10$ s, and thus we had to adjust the $\mathbf{Q}$ covariance matrix $q = \frac{10}{30}3.3 \times 10^{-5} = 1.1 \times 10^{-5}$ (as we assume that $q \propto t_s$). Because of NigeriaSat's higher altitude, it has a longer orbital period, and subsequently we adjusted $t_{act,new} = \frac{P_{NigeriaSat}}{P_{SumbandilaSat}} t_{act,old} = \frac{98.77}{94.62} \times 7 = 7.31$ min and $t_{int,new} = \frac{P_{NigeriaSat}}{P_{SumbandilaSat}} t_{int,old} = \frac{98.77}{94.62} \times 75 = 78.3$ min. The position error of the EKF when the NigeriaSat GPS data is used as input is plotted in Figure 5.23. Upon inspection we found that the GPS receiver on NigeriaSat provided very noisy velocity measurements with an offset in the normal direction. Irregular position measurement fluctuations are also visible when we are propagating only (see left plot in Figure 5.23 just after 100 minutes, just prior to 200 minutes, etc). These are the reasons for the bad results obtained here. As this system will most likely be used in conjunction with a more accurate GPS receiver (see Appendix B.9), we expect that the performance will be more in line with that of the SumbandilaSat simulation.

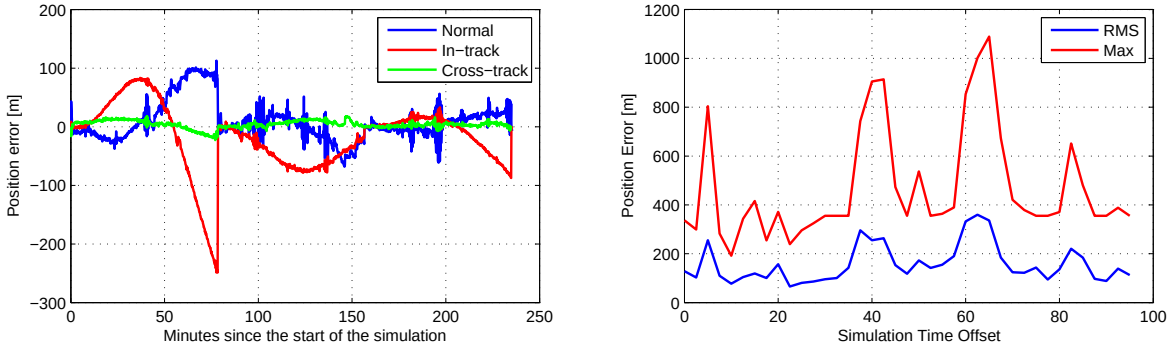


Figure 5.23 – EKF position error with NigeriaSat GPS dataset as input.

5.10 EKF Power and Computational Budget

Using a TTFF range of 1 to 2 minutes (see Appendix B) in Equation 1.2, the EKF system's power consumption factor should be about:

$$\eta_{P,EKF} = \frac{t_{act} + \text{TTFF}}{t_{int}} = \begin{cases} \frac{7+1}{75} = 0.107 & \text{minimum} \\ \frac{7+2}{75} = 0.12 & \text{maximum} \end{cases} \tag{5.21}$$

Using a 260 minute simulation with a sample rate of 30 s, the average execution time was 9 ms. Assuming that this execution time can be scaled to longer simulations, we can expect a 24 hour simulation to take 49.8 ms to complete. Thus $\eta_{comp,EKF} \approx \frac{49.8}{16} = 3.12$.

Chapter 6

Conclusion

6.1 Summary

Almost every LEO satellite mission requires on-board knowledge of position that is accurate. Whilst GPS receivers are widely used on microsatellites (and larger satellites), CubeSats of the traditional sizes cannot afford to keep a GPS receiver powered throughout its entire lifetime. Propagators like SGP4 are the most popular alternative, but these require regular updates from ground stations, and their accuracy deteriorates rapidly with time. These propagators are insufficient for applications that require a sub 500 m accuracy.

We devised an innovative strategy for obtaining on-board accurate knowledge of satellite position without suffering the power consumption burden of a permanently active GPS receiver. Satellite orbit state was estimated using a short interval of GPS receiver measurements, after which propagation algorithms predicted satellite states for a much longer period.

A theoretical study of astrodynamics, specifically applied to LEO satellites, was first done. This not only required an in-depth investigation into the mechanics of perturbations, but also the understanding of exact timekeeping and coordinate frames.

Two solutions, based on different orbit propagating techniques, were developed. The propagation methods, state representations and estimation techniques of these solutions are outlined in Table 6.1.

Table 6.1 – Summary of solutions developed in this thesis.

	aSGP4	EKF
Propagator	SGP4	RK4 + orbital dynamics model
Propagator type	Analytical	Numerical integration
Satellite state representation	Classical orbital elements + Δt	Cartesian position and velocity vectors
State estimation	IIR/averaging of difference	EKF

These two techniques were implemented in the C programming language. Using simulations with recorded SumbandilaSat GPS data as input, these methods were refined and optimised for the problem at hand.

6.2 Results and Evaluation

The simulation results of the two new methods, aSGP4 and the EKF, compared to the standard SGP4 and goals set in Chapter 1 are listed in Table 6.2. Note that the position errors are only approximate. The

SGP4 and aSGP4 results will most probably improve if the B^* terms in the TLEs were more accurate. Further testing using datasets of smaller satellites and lower orbits could reveal more about the true nature of these systems.

Table 6.2 – Results of different propagation techniques and the required targets.

Criteria	Goal	SGP4	aSGP4	EKF
η_P	<15 %	≈ 0%	12.5 - 15 %	10.7 - 12 %
η_{comp}	<5	1	4.125	3.12
RMS position error	100 m	3.2 km	200 m	60 m
Max position error	500 m	5.5 km	1 km	300 m

Power consumption and computational load factors of both systems are within the required margins. The position errors of aSGP4 is greater than what we specified for, whilst the EKF produced fairly decent accuracies during the simulations. Besides the quantitative results above, the following qualitative properties should also be considered:

- The aSGP4 technique, although less accurate, has the advantage of being able to propagate to any arbitrary timestamp with only a single command. Both methods will be able to recover from an OBC reset (due to radiation) if they were able to store their last state in flash memory. Because of its long-range propagation abilities, the aSGP4 would be able to provide a navigation solution with a single step. The numerical integrator, however, will have to propagate stepwise from the saved state to when the OBC is fully operational again.
- SGP4 has a substantial flight heritage, and thus a slightly augmented version could be considered more trustworthy than a whole new method.

6.3 Recommendations for Future Research

Before one of these systems are flown on a satellite, a few recommendations must first be investigated:

Datasets

- The first and foremost recommendation is that the system must be tested with more GPS datasets. Testing with orbits of different altitudes, inclinations and eccentricities could reveal more about these systems' nature. Datasets of lower orbit altitudes and smaller satellite sizes will most probably be more susceptible to the unpredictable aerodynamic drag perturbation, which could require more complex drag models.
- Simulating the EKF system with an artificial dataset that had a faster sampling rate exhibited promising results. Testing the system with true GPS datasets having faster sampling rates might prove that we can use a shorter t_{act}, subsequently improving the expected power consumption. This will, however, increase the computational load.
- 24 h simulations might reveal interesting effects due to a more complete ground track. To analyse these, we will have to obtain such datasets. The CHAMP and GRACE satellite orbit information described in Section 5.1 would have been ideal, but we were unable to obtain these.

Adaptive SGP4

- The current IIR and averaging methods for estimating the aSGP4 Δ parameters is not as accurate as the EKF method, but still exhibited significant improvements over SGP4. It is suggested that further research is done into how NORAD determines its TLE parameters. This will provide more insight and might reveal an improved method of estimating the Δ parameters. The possibility of a least squares method or modifying the Kalman Filter to be compatible with SGP4 as propagator should also be investigated. The OD methods using differential correction of Vallado [37; 12] appeared promising. It must be said, though, that any analytical method is prone to position errors greater than a kilometer, which is inherent to the exclusion of complex perturbations.
- The instantaneous Kepler elements obtained from the GPS measurements (like those plotted in Figure 4.3) were very noisy. This is due to the noise on the GPS velocity measurements. We could use an EKF with a very simple force model to first filter the GPS measurements before we calculate the Δ parameters.

EKF

- Montenbruck [13] has suggested introducing empirical accelerations as states in the EKF. These could imitate the effect of unmodelled perturbations, but it seems that this approach should only be considered for extremely precise orbit determination problems.
- Kahr [7] made an interesting suggestion of using the PDOP of the GPS navigation solutions to grade their quality. Each GPS data point was given a weight of $1/\text{PDOP}^2$. We could use this to scale the $\mathbf{R}_k$ (note the k index to indicate discrete time variation) covariance matrix by the inverse square of PDOP prior to each measurement update.

General

- Estimating the impact of aerodynamic drag (i.e. the C_d and B^* coefficients in numerical and in analytical methods, respectively) can mitigate the secular in-track error grow. This might allow us to use a larger t_{int}, reducing the power consumption.
- It is suggested that a more robust method for detecting corrupt GPS measurements should be investigated. The technique discussed by Cho [28] can be a good starting point.
- Analytical methods are very efficient at predicting satellite positions in the remote future to a medium accuracy grade. It is suggested that certain events such as ground passes are predicted using SGP4, whilst precise navigation solutions are obtained using the EKF and numerical integration method (for example when a scientific payload measurement requires a satellite position stamp).

Appendix A

Time Systems and Coordinate Reference Frames

Reference frames have a vital role in all satellite missions. A variety of reference frames exist because they each simplify the perception of certain objectives. Different modules of the satellite function in different coordinate frames: ADCS components usually use the satellite orbital frame, the GPS receiver outputs are in an Earth-fixed frame, and celestial phenomena and navigation estimators are usually defined in an inertial frame. It is therefore crucial to know the difference and correlation between these coordinate systems before analysing any satellite mission aspect.

For the purpose of orbit description, two coordinate systems are markedly useful: the ECI and ECEF systems. When analysing the nature of a propagator's error, the satellite (orbit) reference frame provides very insightful results.

Before we discuss the coordinate frames, an explanation of time systems that are used must be made. For extremely accurate satellite navigation problems, Einstein's general relativity theorem needs to be applied to correlate the time lapse differences of terrestrial events to those occurring in satellites. Contributors to the difference include Earth's rotational speed, latitude of the terrestrial clock and the extreme velocity of satellites. Montenbruck [13] states that these effects are only relevant for problems involving 1 cm accuracies, and thus are not considered in this thesis.

A.1 Time Systems

> *Time is something we intuitively understand extremely well and, consequently, understand poorly in detail.* [14]

Newcomb stated that "the main purpose of time is to define with precision the moment of a phenomenon" [12]. This moment is referred to as the epoch of the event, and is described by a date. To quantify a date, we need an agreed-upon fundamental reference epoch, and count the number of passings of a certain (repeatable) precise time interval.

An accustomed fundamental epoch is the beginning of the Christian era [12]. Four time scales are used today: sidereal, solar (universal), dynamical, and atomic time. While sidereal and solar time scales are based on Earth's rotation, dynamical and atomic time scales are independent.

A.1.1 Universal Time and UTC

Apparent solar time is the interval between successive transits [12]. Because Earth's orbit about the sun is slightly eccentric, the true (apparent) length of each day differs slightly. This is why astronomers changed to the mean solar time in the late 19th century. The U.S. Naval Observatory determines the mean solar time through GMST (Greenwich Mean Sidereal Time) and corrections for Earth's rotation. Universal Time, UT, is defined as the mean solar time at Greenwich.

There are three forms of UT: UT0, UT1 and UT2 [12]. UT0 is derived by using star observations from many ground stations. When UT0 is corrected for polar motion one obtains UT1. UT2 is a seasonal variation corrected version of UT1.

UTC (Coordinated Universal Time), introduced in January 1972, is the most popular time system [12]. It is based on the atomic clock (the SI definition of time) and is designed to follow UT1 within $\pm$ 0.9 s. Due to variations in Earth's rotation, UT1 presents fluctuations. Leap seconds are periodically inserted into UTC to keep up with UT1. This occurs at the end of either June or December. A history of when leap seconds were inserted can be found on the International Earth Rotation and Reference Systems Service (IERS) website [58]. Figure A.1 shows the difference between UT1 and UTC from 1973 to 2006.

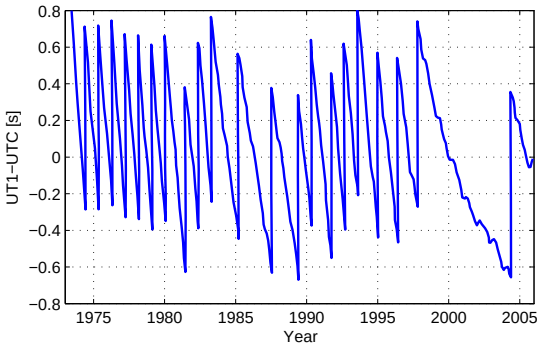


Figure A.1 – Difference between UT1 and UTC time systems.

A.1.2 Julian Date and J2000

The Julian date is a central concept in the astrodynamics industry [12]. Its epoch is at 12:00 1 January 4713 B.C. (UT).

The lengths of the solar and lunar cycles were known to a quite reasonable accuracy by the ancients [59]. Many societies based their time systems on either lunar or solar periods, or even a combination. These systems would even change as new rulers emerged. Some years had extra months inserted to keep up with the actual solar year, and serious confusion around calendar systems was present at the time. To resolve this confusion, Julius Caesar inaugurated the Julian Calendar (drafted by Sosigenes) on 1 January 45 BC. This calendar was altered by Augustus during his reign.

Even though this system fixed the confusion and was used by almost the entire known world of the time, each year was actually 11 minutes, 4 seconds longer than a real solar year [59]. Pope Gregory XIII corrected this in 1582, by which time the error had already accumulated to 10 days. He proclaimed that 15 October would immediately succeed 4 October that year, and that leap years ending in 00 will no longer be leap years, unless divisible by 400. This is known as the Gregorian calender, which is used in civil society today.

As England was still using the Julian calendar until 1752, the world was again divided in terms of calendar systems [59]. As no one could agree upon definitions for months and years, Joseph Justus Scaliger invented the Julian period, a system that only depended on an epoch and the number of days that have passed. The system is named after Scaliger's father, Julius Caesar Scaliger, and not Julius Caesar the Roman emperor. He combined the solar (activity) cycle (28 years), Metonic cycle (19 years), and the Roman taxation cycle of indication (15 years), to form a period of 7980 years. The epoch of 12:00 1 January 4713 B.C. (UT) was chosen as the start of Julian day 0 because it is the closest past year on which all three of these systems coincided.

Meeus [60; 12] describes a formula for the calculation of a Julian date from a calendar date (Algorithm A.1). Notice how January and February are represented as the 13^{th} and 14^{th} months of the previous year.

Algorithm A.1 cal2JD

Input = $[yr, mn, d, h, min, s]$ = Gregorian calendar year, month, day, hour, minute and second

if $mn <= 2$ **then**

$yr = yr - 1$

$mn = mn + 12$

end if

$$B = 2 - \text{int}\left(\tfrac{y}{100}\right) + \text{int}\left(\frac{\text{int}\left(\frac{y}{100}\right)}{4}\right)$$

$$C = \frac{1}{24}\left(\frac{1}{60}\left(\frac{s}{60} + min\right) + h\right)$$

$$JD = \text{int}(365.25(yr + 4716)) + \text{int}(30.6001(mn + 1)) - 1524.5 + d + B + C$$

For example, 00:00 1 January 2015 corresponds to Julian date 2457023.5. We sometimes apply the Julian date concept to different epochs. A popular example is the J2000 date, which, as its name suggests, has its epoch on 12:00 1 January 2000 (Julian date 2451545). To obtain the J2000 day and fraction of day is trivial:

$$J2000 = JD - 2451545 \tag{A.1}$$

Sometimes it is more convenient to describe some phenomenon in terms of fractions of Julian centuries. The fraction of Julian centuries since 1 January 2000, T_{J2000}, is found with the following formula [61]:

$$T_{J2000} = J2000/36525 = (JD - 2451545)/36525 \tag{A.2}$$

A.1.3 GPS Time

GPS time is the time scale to which GPS signals are referenced [62]. GPS time is steered to be within 1 µs of UTC, ignoring leap seconds. GPS time and UTC were exactly equal in 1980, but leap seconds have since been inserted into UTC, effectively letting these two drift apart in integer second intervals. As of 1 July 2015, GPS time is ahead by 17 s [63]. Note that TAI and GPS time is always out by 19 s (as both neglect leap seconds).

GPS time is usually stated in GPS week number (WN) and second of week (SOW) [62]. GPS WN is a modulo-1024 week count which started at midnight on the night of 5 Jan / morning of 6 Jan, 1980. A new week starts at midnight Saturday night / Sunday morning, during which the SOW counter will roll over from 604799 to 0.

A *week rollover* is an event which occurs every 1024 weeks (just short of 20 years) when the WN restarts at 0. GPS modules must be able to handle this, and the number of rollovers must be taken into account when converting GPS time to other formats. At the time of publication, there has only been one GPS week rollover, which occurred on 23:59:47 UTC Saturday, 21st August 1999. Algorithm A.2 shows how GPS time can be converted to Julian date.

Algorithm A.2 GPStime2JD

Input = [WN, SOW, $n_{rollover}$] = current week number, second of week, number of week rollovers

$n_{week} = \text{WN} + 1024 \times n_{rollover}$

$\text{JD} = \text{cal2JD}(6 \text{ Jan } 1980\ 00{:}00{:}00) + 7n_{week} + \text{SOW}/86400$

A.1.4 TLE Epoch

TLE set epochs are given in a year and DoY (day of year) format [10]. Only the last two digits of every year are provided, and thus entries 57 - 99 correspond to 1957 - 1999, whilst entries 00 - 56 represent 2000 - 2056. An update of all TLE interpretation programs and/or TLE format will therefore be required within the next 40 years. Equation A.3 shows how simple the conversion is from TLE epoch to JD.

$$\begin{aligned} year &= \begin{cases} 1900 + yr & \text{if } yr \geq 57 \\ 2000 + yr & \text{if } yr \leq 56 \end{cases} \\ \text{JD} &= \text{cal2JD}(year, 1, 1, 0, 0, 0) + \text{DoY} \end{aligned} \tag{A.3}$$

where yr is the two digits of the year.

A.2 Earth-based Coordinate Frames

A.2.1 ECI Frame

The equatorial and ecliptic planes are separated by an angle called the obliquity of the ecliptic [12]. The line where these two planes intersect is called the line of nodes. The sun crosses the equatorial plane twice a year: one where the sun is ascending (vernal equinox, around March 21) and one where it is descending (autumnal equinox, around September 23). These events correspond to when the Earth experiences equal times of day and night. The formal definition of the vernal equinox is that it occurs when the sun declination is zero, which differs from the description above as we often use the mean, instead of the true, path of the Sun to determine the ecliptic plane. Vernal equinox is often referred to as the first point of Aries and designated with an ♈. This is because the vernal equinox pointed to this constellation in Christ's lifetime. Due to precession (discussed in Appendix A.2.3.1) the vernal equinox direction has shifted, and today it is directed more towards the Pisces constellation.

The Earth Centred Inertial (ECI) coordinate frame explains itself in its name. It has its origin at the centre of the mass of the Earth, and it is inertially fixed to the distant stars. Its axes are defined as following: X-axis towards the vernal equinox, Z-axis in the direction of the Earth North pole, and the Y-axis right-handedly orthogonal to these two. Figure A.2 illustrates the ECI frame definition. The ECI X-Y plane thus, by definition, corresponds to the equatorial plane.

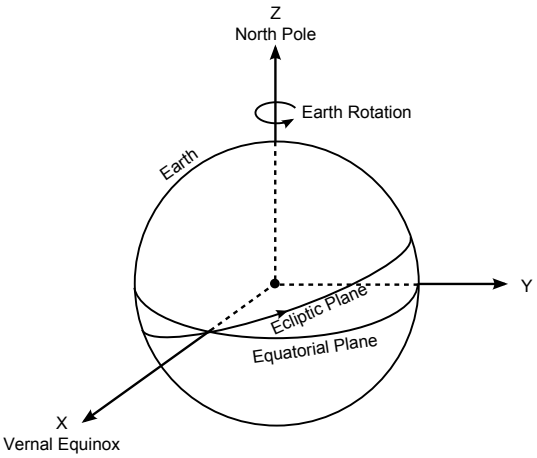


Figure A.2 – Illustration of the ECI frame.

An inertial system is defined as non-rotating and non-accelerating [64]. As the Earth orbits the sun, it is constantly experiencing an acceleration towards the sun, and thus any system that has its origin at Earth's centre is, by strict definition, not truly inertial (the same argument can be made for the Sun rotating about the Milky Way). However, as Vallado [12] states, we can assume this to be a quasi-inertial frame when we use an appropriate epoch time to fix the vernal equinox and North pole (these move throughout time, as discussed in Section A.2.3). A quasi-inertial reference frame is a system in which the small non-inertial effects are either too small to measure, or are negligible during the period of interest. For the purpose of this thesis, a quasi-inertial frame will suffice.

A.2.2 ECEF Frame

The Earth Centred Earth Fixed (ECEF) coordinate frame is fixed to Earth's surface. It shares the same origin as the ECI frame, namely Earth's centre of mass. It thus rotates around Earth's North pole. Inherently, this frame is highly non-inertial. It has the Z-axis pointed towards the Earth North pole, the X-axis directed to the Greenwich Meridian, and the Y-axis defined right handedly orthogonal relative to these two. This reference frame is useful for navigation on Earth's surface, and is thus the output of most GPS modules. It is also used for tasks that require pointing towards targets on Earth, such as communication or Earth imaging.

A.2.3 Transformation Between ECEF and ECI Frames

As discussed in Chapter 5, the satellite state is defined in the ECI frame, and thus the GPS data will need to be transformed to the correct reference frame. Hashida [61] describes the transformation (from ECI to ECEF) in four steps: precession, nutation, polar motion, and rotation.

For convenience, we first define the three principle axis rotation matrices as:

$$\mathbf{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & \sin\theta \\ 0 & -\sin\theta & \cos\theta \end{bmatrix} \quad \mathbf{R}_y(\theta) = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \quad \mathbf{R}_z(\theta) = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{A.4}$$

A.2.3.1 Precession

Precession is the smooth long periodic change of Earth's equatorial plane about its orbital pole [14]. It can be thought of as the "wobbling" that a spinning top undergoes. The main cause for this effect is the

gravity forces of the other planets in our solar system. In addition to affecting the Z-axis of the ECI frame, it also alters the vernal equinox direction. The result of this process is that the Vernal equinox changes its direction relative to the fixed stars at an average rate of about 0.014°/year. Precession can be accounted for by means of a rotation matrix $\mathbf{P}$ defined in equation A.5 [61]:

$$\begin{aligned}
\theta_p &= 2004.3109T_{J2000} - 0.42665T_{J2000}^2 - 0.041833T_{J2000}^3 \\
z_p &= 2306.2181T_{J2000} + 1.09468T_{J2000}^2 + 0.018203T_{J2000}^3 \\
\zeta_p &= 2306.2181T_{J2000} + 0.30188T_{J2000}^2 + 0.017998T_{J2000}^3 \\
\mathbf{P} &= \mathbf{R}_z(-z_p)\mathbf{R}_y(\theta_p)\mathbf{R}_z(-\zeta_p)
\end{aligned} \tag{A.5}$$

A.2.3.2 Nutation

The precise movement of Earth's North Pole is not a smooth phenomenon as the precession equations would describe it. Nutation refers to this phenomenon's shorter periodic effects, which are primarily caused by the Moon [12]. To account for Nutation effects, a series expansion is required. This method can be and immense process if a full model is used. For example, the IERS 1996 Theory of Precession and Nutation has 263 terms.

A.2.3.3 Polar Motion

According to [61], the polar motion has a 9m variation. It is assumed that this variation is a very slow process, thus only introducing an offset and not affecting the dynamics.

A.2.3.4 Rotation

The most substantial (and obvious) transformation between ECEF and ECI is the rotation of the Earth about its own axis. The transformations, shown in Equation A.6, differ for position and velocity vectors [26].

$$\begin{aligned}
\theta &= \theta_{ECEF\rightarrow ECI} = -\text{GST} \times \frac{\pi}{12} \\
\begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECI} &= \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_{ECEF} \\
\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix}_{ECI} &= \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{x} - y \times \omega_e \\ \dot{y} + x \times \omega_e \\ \dot{z} \end{bmatrix}_{ECEF}
\end{aligned} \tag{A.6}$$

where GST (Greenwich Sidereal Time) is the angle between the (Greenwich) prime meridian and the vernal equinox (in hours). GMST, Greenwich Mean Sidereal Time, an average of the true angle, can be used for GST. It can be roughly approximated using Equation A.7 [65].

$$\text{GMST} = 18.697374588 + 24.06570982441908 \times J2000 \tag{A.7}$$

A better approximation of GMST is given in Equation A.8 [65].

$$\text{GMST} = 6.697374558 + 0.06570982441908D_0 + 1.00273790935H + 0.000026T^2 \tag{A.8}$$

where D_0 is the integer number of days since J2000, H is the hours past the previous midnight, and T is the fractional number of centuries since J2000.

A more accurate angle for Earth's rotation is the Greenwich Apparent Sidereal Time (which is measured to the true equator). Equation A.9 shows how du Plessis [66] summarised the formula of Seidelmann [67] to determine GAST. The US Defence Mapping Agency released a similar approach in a report [68].

$$
\begin{aligned}
\epsilon_m &= (84381.448 - 46.815T - 0.00059T^2 + 0.001813T^3) \times \frac{\pi}{43200} \\
L &= (280.4665 + 36000.7698T) \times \frac{\pi}{180} \\
\Delta L &= (218.3165 + 481267.8813T) \times \frac{\pi}{180} \\
\Omega &= (125.04452 - 1934.136261T) \times \frac{\pi}{180} \\
\Delta\Psi &= (-17.20\sin(\Omega) - 1.32\sin(2L) - 0.23\sin(\Delta L) + 0.21\sin(2\Omega)) \times \frac{\pi}{648000} \\
\Delta\epsilon &= (9.20\cos(\Omega) + 0.57\cos(2L) + 0.10\cos(\Delta L) - 0.09\cos(2\Omega)) \times \frac{\pi}{648000} \\
\text{GAST} &= \text{GMST} + \Delta\Psi\cos(\epsilon_m + \Delta\epsilon)
\end{aligned}
\tag{A.9}
$$

A.2.3.5 SumbandilaSat GPS Data

The provided raw GPS data is in the ECEF reference frame, and needs to be converted to the ECI frame.

We will test our conversion methods by transforming the GPS data. Altitude and velocity magnitude can be obtained from Equation A.10.

$$
\begin{aligned}
\text{Altitude} &= \sqrt{x^2 + y^2 + z^2} - R_e \\
|\boldsymbol{v}| &= \sqrt{\dot{x}^2 + \dot{y}^2 + \dot{z}^2}
\end{aligned}
\tag{A.10}
$$

The ECEF and ECI corresponding altitudes and velocity magnitudes are compared in Figure A.3. It is clear that the rotation process does not affect the altitude, but the velocity magnitude is markedly affected due to the Coriolis Effect. The oscillation of the altitude (and velocity) in the ECI frame suggests that SumbandilaSat was in an eccentric orbit at this stage. At around 125 minutes, we see some anomalies (spikes). These are due to corrupt GPS samples.

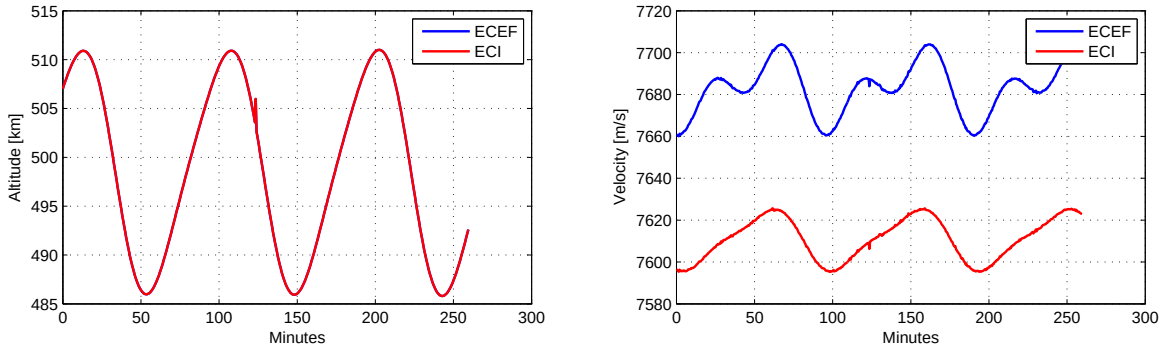


Figure A.3 – Comparison of ECEF and ECI representations of SumbandilaSat altitude and velocity magnitude.

A.3 Satellite Orbit Coordinate Frames

We will make use of the Satellite coordinate frame to obtain a better understanding of the nature of the position errors that we will encounter. Vallado describes two ways to define such coordinate systems, the RSW (radial, along-track and cross-track) and NTW (normal, in-track and cross-track) systems. These

two might seem alike, but in fact they are only identical when the satellite involved is in a perfectly circular orbit. The RSW produces a true radial vector, but its along-track vector does not depict the direction the satellite is moving in, while NTW produces a true in-track vector, but its normal vector does not represent the radius from the centre of Earth. The two frames are illustrated in Figure A.4.

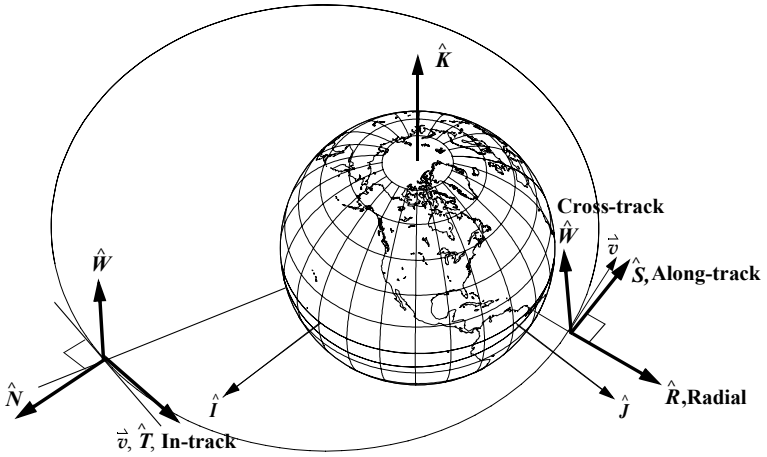


Figure A.4 – Satellite (orbit) coordinate frames. Derived from Vallado [12].

The transformations from ECI to RSW and NTW frames are described by Vallado [12]:

$$\begin{aligned} \mathbf{R} &= \frac{\mathbf{r}}{|\mathbf{r}|} \qquad \mathbf{W} = \frac{\mathbf{r}\times\mathbf{v}}{|\mathbf{r}\times\mathbf{v}|} \qquad \mathbf{S} = \mathbf{W}\times\mathbf{R} \\ \mathbf{r}_{RSW} &= \begin{bmatrix} \mathbf{R} \\ \mathbf{S} \\ \mathbf{W} \end{bmatrix} \mathbf{r}_{IJK} \end{aligned} \tag{A.11}$$

$$\begin{aligned} \mathbf{T} &= \frac{\mathbf{v}}{|\mathbf{v}|} \qquad \mathbf{W} = \frac{\mathbf{r}\times\mathbf{v}}{|\mathbf{r}\times\mathbf{v}|} \qquad \mathbf{N} = \mathbf{T}\times\mathbf{W} \\ \mathbf{r}_{NTW} &= \begin{bmatrix} \mathbf{N} \\ \mathbf{T} \\ \mathbf{W} \end{bmatrix} \mathbf{r}_{IJK} \end{aligned} \tag{A.12}$$

where the matrices are referred to the RSW and NTW DCM's, respectively. Note that DCM's are orthogonal, and thus their inverse is equal to their transpose:

$$\begin{bmatrix} \mathbf{R} \\ \mathbf{S} \\ \mathbf{W} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{R} \\ \mathbf{S} \\ \mathbf{W} \end{bmatrix}^{T} \qquad \begin{bmatrix} \mathbf{N} \\ \mathbf{T} \\ \mathbf{W} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{N} \\ \mathbf{T} \\ \mathbf{W} \end{bmatrix}^{T} \tag{A.13}$$

Appendix B

GPS

The Global Positioning System (GPS) is a space-based navigation system developed under the U.S. Department of Defence's NAVigation Satellite Timing And Ranging system (NAVSTAR) program. Accurate position, velocity and time information is provided to an unlimited amount of users world-wide and in all weather conditions. It consists of a constellation of 24 or more active satellites, with ground segments uploading information to these satellites. These satellites emit radio waves in the L-band, which a receiver can use passively to estimate its navigation solution [69].

The satellites are spread in six 55° inclined orbits, separated from each other by 60° in RAAN. Their orbits are nearly circular and have radii of 26560 km, and thus an orbital period of half a sidereal day. This constellation guarantees that four or more satellites with a good geometrical relationship will be visible at any location and time on Earth [62].

For comprehensive discussions on GPS, consult the works of Grewal [62], Kaplan [70] and the NAVSTAR GPS User Equipment Introduction [69].

B.1 Brief History

The U.S. Navy Navigation Satellite System (often referred to as Transit), which became operational in 1964, was the first U.S. space-based navigation system. It was a five satellite constellation, and provided two-dimensional positioning. The frequency of which a position fix could be obtained depended on the receiver's latitude, and this typically varied between 30 and 110 minutes. Due to the computer technology of at the time, it required an initial estimation of user position and took 10 to 15 minutes to perform its calculations. This made it acceptable for shipboard navigation, but it was inadequate for vehicles travelling at higher velocities [70].

Since the Transit system proved to be a success, the Department of Defense (DOD), National Aeronautics and Space Administration (NASA) and the Department of Transportation (DOT) joined forces to improve on this system. Their objective was to develop a system with the following characteristics:

- Global coverage
- Continuous and all weather operation
- Capable of serving receivers moving at higher velocities
- Highly accurate

GPS was partly operational in December 1993, and became fully operational by early 1995. GPS offers two services: Standard Position Service (SPS) for civilian use, and Precise Positioning Service (PPS) for U.S. military use. Access restriction to the PPS service is achieved by cryptographic techniques called antispoofing and selective availability (SA). By dithering the transmitted GPS clock and data parameters, SA deliberately degraded accuracy, unless the receiver had a decryption code. As of May 1 2000, SA was discontinued. The SPS service is freely available to any user around the world, and it is specified to have 95% accuracies of 13 m horizontally, 22 m vertically and 40 ns in clock time, but it will typically perform better than this [70].

B.2 Signal Description

Each GPS satellite transmits two spread spectrum L-band signals: L_1 and L_2 at 1575.42 MHz and 1227.6 MHz carrier frequencies, respectively. Basic GPS receivers make use of only the L_1 signal, but the dual frequency system can yield more accurate solutions. This is because the ionosphere induces an additional delay in propagation, which is roughly proportional to the inverse of the square of signal frequency, and can thus be approximated by this dual frequency system. Our GPS module, however, will be spaceborne, and thus will not be exposed to the full delay caused by the ionosphere. For most satellite applications this compensation is needless [62].

The L_1 signal uses binary phase-shift keying (BPSK) which are modulated by two pseudorandom noise (PRN) codes: the so-called C/A-code (coarse acquisition) and P-code (precision code), while the L_2 signal is modulated by the P-code only. PRN codes enable the receiver to access multiple GPS satellites' signals, which are all on the same carrier frequency. Each GPS satellite has unique C/A and P-codes. The C/A code is used for rapid satellite signal acquisition and it is thus a relatively short and coarse-grained code. It has 1023 chips (increments until it gets repeated) with a chip-rate of 1.023 MHz. The full P-code has a 259 day length, however, each GPS satellite uses a only a unique 7 day portion of this full code. At a 10.23 MHz chipping rate, it has a bandwidth 10 times wider than the C/A-code. This greatly reduces the errors due to multipath and receiver noise. Figure B.1 illustrates the structure of the two GPS signals transmitted on the L_1 frequency [62].

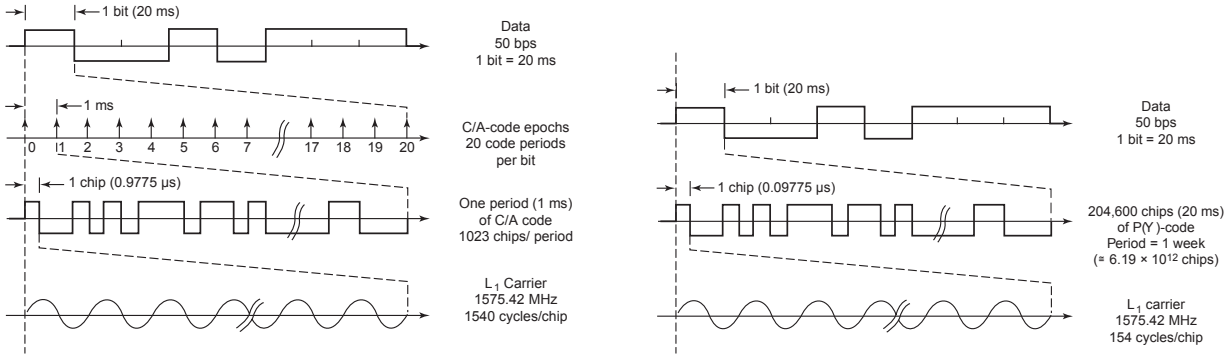


Figure B.1 – Structure of GPS L_1 in-phase (left) and quadra-phase (right) signals. Obtained from [62].

B.3 Navigation Message

The GPS navigational data stream is transmitted at a 50 baud rate. This contains an almanac for all the GPS satellites, the broadcasting satellite's ephemeris, signal timing and satellite clock offsets, as well

as information about signal propagation delays (for single frequency receivers). The almanac enables the user to approximate the location of every satellite in the GPS constellation. Though this information is not accurate enough for determining the user's position, it is used to determine which satellites should be visible to the user, so that those satellites could be searched for when the module is switched on. It is also used to approximate the signals' expected Doppler shift. The satellite ephemeris data is much more accurate and is used to transform a GPS satellites' signal propagation time into a pseudorange. While the almanac data is valid for several weeks, the satellite ephemeris data can only be used for a few hours [62].

A complete message contains 25 frames, each divided into five 300-bit subframes. At 50 bps a subframe would thus take 6 s to transmit, a frame 30 s and a full 25 frame message 12.5 minutes. Subframe 1 contains the clock correction, and 2 and 3 the satellite ephemeris data. The latest version of these frames can thus be obtained every 30 s. Subframes 4 and 5 are concatenated to construct a 25 page message (the reason why a full GPS message requires 25 frames), each frame providing a page. The message contains the almanac, ionospheric delay data and other information. The structure of a frame is illustrated in Figure B.2. TLM and HOW represent telemetry and hand-over words, respectively [62].

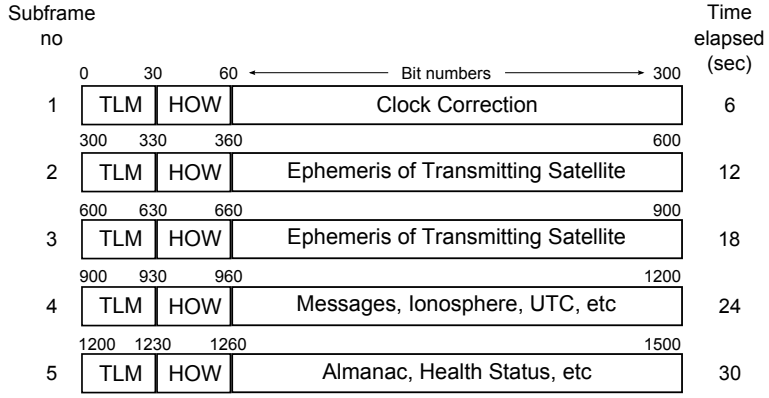


Figure B.2 – Structure of GPS navigation message, reproduced from [62].

The almanac and ephemeris contain virtually the same parameters; the data of the latter is just much more precise. Table B.1 summarizes the contents of a GPS ephemeris. With these parameters, Algorithm B.1 can be used to calculate a GPS satellite's position in the ECEF reference frame. Here, t is in GPS time, corrected for propagation time (range to satellite / speed of light). μ and ω_e are the Earth's universal gravitational parameter and rotation rate [62].

Table B.1 – Ephemeris data components [62; 70].

Symbol	Description	Units
M_0	Mean anomaly at reference time	semicircle
Δn	Mean motion difference from computed value	semicircle/s
e	Eccentricity	-
$\sqrt{a}$	Square root of semimajor axis	$\mathrm{m}^{\frac{1}{2}}$
Ω_0	Longitude of ascending node at weekly epoch	semicircle
i_0	Inclination at reference time	semicircle
ω	Argument of Perigee	semicircle
$\dot{\Omega}$	Rate of longitude of ascending node	semicircle/s
IDOT	Rate of inclination	semicircle/s

$C_{\mu c}$	Cosine harmonic correction term to the argument of latitude	rad
$C_{\mu s}$	Sine harmonic correction term to the argument of latitude	rad
C_{rc}	Cosine harmonic correction term to the orbit radius	m
C_{rs}	Sine harmonic correction term to the orbit radius	m
C_{ic}	Cosine harmonic correction term to the inclination	rad
C_{is}	Sine harmonic correction term to the inclination	rad
t_{0e}	Ephemeris reference time	s
IODE	Issue of data, ephemeris	-

Algorithm B.1 Computing GPS satellite position from ephemeris data [62; 70; 71].

$a = (\sqrt{a})^2$	Semimajor axis
$n_0 = \sqrt{\mu/a^3}$	Computed mean motion
$t_k = t - t_{0e}$	Time from ephemeris epoch
$n = n_0 + \Delta n$	Corrected mean motion
$M_k = M_0 + nt_k$	Mean anomaly
$M_k = E_k - e\sin(E_k)$	Solve Kepler's equation iteratively for E_k
$\dot{E}_k = \dfrac{n}{1 - e\cos(E_k)}$	First derivative of eccentric anomaly
$v_k = \arctan 2\left(\sqrt{1-e^2}\sin E_k, \cos E_k - e\right)$	True anomaly
$\dot{v}_k = \dfrac{\dot{E}_k(1 + e\cos v_k)\sin E_k}{\sin v_k(1 - e\cos E_k)}$	First derivative of true anomaly
$\phi_k = v_k + \omega$	Argument of latitude
$\delta\mu_k = C_{\mu c}\cos 2\phi_k + C_{\mu s}\sin 2\phi_k$	Harmonic correction of argument of latitude
$\delta r_k = C_{rc}\cos 2\phi_k + C_{rs}\sin 2\phi_k$	Harmonic correction of radius
$\delta i_k = C_{ic}\cos 2\phi_k + C_{is}\sin 2\phi_k$	Harmonic correction of inclination
$\mu_k = \phi_k + \delta\mu_k$	Argument of latitude correction
$r_k = a(1 - e\cos E_k) + \delta r_k$	Radius correction
$i_k = i_0 + \delta i_k + (\mathrm{IDOT})t_k$	Inclination correction
$\dot{\mu}_k = \dot{v}_k + 2\dot{v}_k\big(C_{\mu s}\cos(2\mu_k) - C_{\mu c}\sin(2\mu_k)\big)$	First derivative of μ_k
$\dot{r}_k = \dfrac{aen\sin(E_k)}{1 - e\cos(E_k)} + 2\dot{v}_k\big(C_{rs}\cos(2\mu_k) - C_{rc}\sin(2\mu_k)\big)$	First derivative of r_k
$\dot{i}_k = \mathrm{IDOT} + 2\dot{v}_k\big(C_{is}\cos(2\mu_k) - C_{ic}\sin(2\mu_k)\big)$	First derivative of i_k
$x'_k = r_k\cos(\mu_k)$	X coordinate in orbit plane
$y'_k = r_k\sin(\mu_k)$	Y coordinate in orbit plane
$\dot{x}'_k = \dot{r}_k\cos(\mu_k) - y'_k\dot{\mu}_k$	X velocity in orbit plane
$\dot{y}'_k = \dot{r}_k\sin(\mu_k) + x'_k\dot{\mu}_k$	Y velocity in orbit plane
$\Omega_k = \Omega_0 + (\dot{\Omega} - \omega_e)t_k - \omega_e t_{0e}$	Longitude of ascending node correction
$\dot{\Omega}_k = \dot{\Omega} - \omega_e$	First derivative of Ω_k
$x_k = x'_k\cos\Omega_k - y'_k\cos i_k\sin\Omega_k$	ECEF X coordinate
$y_k = x'_k\sin\Omega_k - y'_k\cos i_k\cos\Omega_k$	ECEF Y coordinate
$z_k = y'_k\sin i_k$	ECEF Z coordinate
$\dot{x}_k = (\dot{x}'_k - y_k\dot{\Omega}_k\cos i_k)\cos\Omega_k - (x'_k\dot{\Omega}_k + \dot{y}'_k\cos i_k - y'_k\dot{i}_k\sin i_k)\sin\Omega_k$	ECEF X velocity
$\dot{y}_k = (\dot{x}'_k - y_k\dot{\Omega}_k\cos i_k)\sin\Omega_k + (x'_k\dot{\Omega}_k + \dot{y}'_k\cos i_k - y'_k\dot{i}_k\sin i_k)\cos\Omega_k$	ECEF Y velocity
$\dot{z}_k = \dot{y}'_k\sin i_k + y'_k\dot{i}_k\cos i_k$	ECEF Z velocity

B.4 Calculating User Position and Velocity

We will discuss an elementary process to calculate the user position and velocity with four GPS satellites. Note that for most cases, more than four satellites are visible to the GPS module, and then least squares

or Kalman filtering methods must be implemented. As this thesis is based on an already functioning GPS module, we will not discuss the latter two methods. Grewal [62] and Kaplan [70] explain recursive methods for position and velocity determination.

B.4.1 Pseudorange Measurement

A pseudorange is the apparent distance between the receiver antenna and the GPS satellite, and is calculated from the time between transmission and reception and the speed of signal propagation (speed of light). This ranging method requires a very accurate synchronization of the GPS satellite and user clocks. GPS satellites have very precise and stable atomic clocks on-board, but receivers don't, and thus we introduce a user clock offset parameter as an additional unknown. Another difficulty to overcome is the fact that Einstein's theory of general relativity needs to be taken into account when working with these fine time measurements and the velocities associated with satellites, however the explanation of this is beyond the scope of this thesis.

Most GPS modules use the C/A-code to make pseudorange measurements [62]. The "state of a C/A-code" is the number and fraction of chips that have occurred since the previous C/A-code epoch (thus a real number between 0 and 1023). Figure B.3 illustrates the concept. The pseudorange to the i'th satellite is:

$$\rho_{r_i} = c\left(t_{rcv} - t_{e_i} - \frac{X_i}{c_r}\right) \tag{B.1}$$

where X_i is the state of the i^{th} satellite's C/A-code, t_{rcv} and t_{e_i} are the times of the module reception and the GPS satellite i's transmission, respectively, and c_r is the C/A-code chipping rate of 1.023×10^6 chips/s.

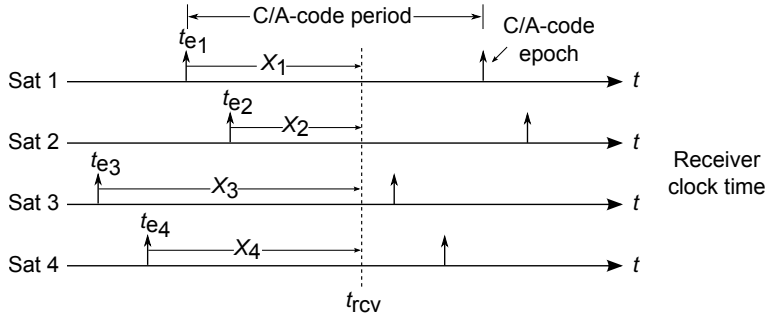


Figure B.3 – Determining pseudoranges from C/A-code. Reproduced from [62].

A much higher pseudorange accuracy can be obtained when the carrier phase is used. Because of the very short carrier signal period, this method's accuracy is typically 10-100 times more accurate than using code delays. Pseudorangerates can be determined using frequency shift (Doppler Effect) measurements. Refer to Grewal [62] and Kaplan [70] for more comprehensive discussions.

B.4.2 User Position

In this discussion we assume that our knowledge of GPS satellite positions and the pseudoranges are correct. The following equation should be apparent [62]:

$$\rho_r = \sqrt{(x-X)^2 + (y-Y)^2 + (z-Z)^2} \tag{B.2}$$

where

$$\begin{aligned} \rho_r &= \text{known pseudorange} \\ x, y, z &= \text{known GPS satellite coordinates} \\ X, Y, Z &= \text{unknown user coordinates} \end{aligned}$$

Grewal [62] shows a simple closed form solution to finding user position, but it is only valid for terrestrial users. We will show a simple generalized closed form solution that can be used for spaceborne users. Squaring Equation B.2 and introducing a clock bias correction term Crr (to compensate for the difference between user and GPS time) we obtain:

$$\begin{aligned} \rho_{r_i}^2 &= (x_i - X)^2 + (y_i - Y)^2 + (z_i - Z)^2 + \text{Crr} \\ &= x_i^2 + y_i^2 + z_i^2 - 2Xx_i - 2Yy_i - 2Zz_i + X^2 + Y^2 + Z^2 + \text{Crr} \end{aligned} \tag{B.3}$$

where $i = 1, 2, 3, 4$ denotes the indices of four visible GPS satellites. Note that X, Y, Z and Crr are the only unknowns. Subtracting the equations of $i = 2, 3, 4$ by that of $i = 1$, we obtain:

$$\begin{aligned} \rho_{r_j}^2 - \rho_{r_1}^2 &= (x_j^2 + y_j^2 + z_j^2) - (x_1^2 + y_1^2 + z_1^2) - 2X(x_j - x_1) - 2Y(y_j - y_1) - 2Z(z_j - z_1) \\ &= r_j^2 - r_1^2 - 2X(x_j - x_1) - 2Y(y_j - y_1) - 2Z(z_j - z_1) \end{aligned} \tag{B.4}$$

where $j = 2, 3, 4$ and $r_j = \sqrt{x_j^2 + y_j^2 + z_j^2}$. We can write these three equations in matrix form:

$$\begin{bmatrix} \rho_{r_2}^2 - \rho_{r_1}^2 - r_2^2 + r_1^2 \\ \rho_{r_3}^2 - \rho_{r_1}^2 - r_3^2 + r_1^2 \\ \rho_{r_4}^2 - \rho_{r_1}^2 - r_4^2 + r_1^2 \end{bmatrix} = \begin{bmatrix} -2(x_2 - x_1) & -2(y_2 - y_1) & -2(z_2 - z_1) \\ -2(x_3 - x_1) & -2(y_3 - y_1) & -2(z_3 - z_1) \\ -2(x_4 - x_1) & -2(y_4 - y_1) & -2(z_4 - z_1) \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{B.5}$$

$$\mathbf{R} = \mathbf{M}\mathbf{U}_r$$

and then obtaining user position $\mathbf{U}$ becomes simple linear algebra:

$$\mathbf{U}_r = \mathbf{M}^{-1}\mathbf{R} \tag{B.6}$$

Crr can be found by plugging the user position values into any version of Equation B.3.

B.4.3 User Velocity

Unlike Grewal's [62] closed form position determination, his method for velocity does not require the user to be on Earth's surface. Differentiating Equation B.2, we obtain:

$$\dot{\rho}_r = \frac{(\dot{x} - \dot{X})(x - X) + (\dot{y} - \dot{Y})(y - Y) + (\dot{z} - \dot{Z})(z - Z)}{\rho_r} \tag{B.7}$$

where $\dot{X}, \dot{Y}, \dot{Z}$, the user velocity components, are the only unknowns. We assume that the pseudorange rate and GPS satellite velocities are known and that the user position is calculated from the work above. Rearrange to get:

$$-\rho_{r_i}\dot{\rho}_{r_i} + \dot{x}_i(x_i - X) + \dot{y}_i(y_i - Y) + \dot{z}_i(z_i - Z) = \dot{X}(x_i - X) + \dot{Y}(y_i - Y) + \dot{Z}(z_i - Z) \tag{B.8}$$

for three satellites ($i = 1, 2, 3$). In matrix form we get:

$$\begin{bmatrix} -\rho_{r_1}\dot{\rho}_{r_1} + \dot{x}_1(x_1 - X) + \dot{y}_1(y_1 - Y) + \dot{z}_1(z_1 - Z) \\ -\rho_{r_2}\dot{\rho}_{r_2} + \dot{x}_2(x_2 - X) + \dot{y}_2(y_2 - Y) + \dot{z}_2(z_2 - Z) \\ -\rho_{r_3}\dot{\rho}_{r_3} + \dot{x}_3(x_3 - X) + \dot{y}_3(y_3 - Y) + \dot{z}_3(z_3 - Z) \end{bmatrix} = \begin{bmatrix} x_1 - X & y_1 - Y & z_1 - Z \\ x_2 - X & y_2 - Y & z_1 - Z \\ x_3 - X & y_3 - Y & z_1 - Z \end{bmatrix} \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{Y} \end{bmatrix} \tag{B.9}$$

$$\mathbf{D}_R = \mathbf{N}\mathbf{U}_v$$

and finally the user velocity is

$$\mathbf{U}_v = \mathbf{N}^{-1}\mathbf{D}_R \tag{B.10}$$

B.5 Dilution of Precision

In essence, Dilution of Precision (DOP) is a measure (which can be quantified) of navigation solution degradation due to the geometry of the GPS satellites used [62]. Figure B.4 illustrates how GPS satellite geometry affects uncertainty in position estimation (in a 2D example). The dotted lines indicate possible ranging error (due to clock errors) and subsequently the grey area represents the uncertainty of user position. In the left illustration, the two transmitters are located close to each other, yielding a larger grey area, whilst a wider configuration on the right side led to a much more certain solution.

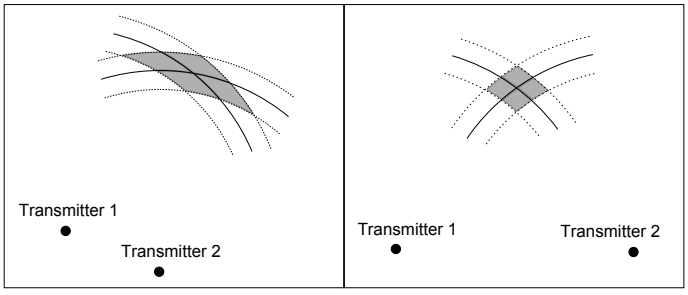


Figure B.4 – 2D DOP concept illustration.

Different types of DOP are used: geometrical (GDOP), time (TDOP), positional (PDOP), horizontal (HDOP) and vertical (VDOP). The latter is especially of importance to spaceborne GPS modules, as erroneous orbital altitude estimation will translate to a large in-track propagation error over time. We will discuss a basic process to calculate GDOP and PDOP, consult Grewal [62] and Langley [72] for a broader discussion. Define matrices $\mathbf{H}$ and $\mathbf{A}$ (covariance):

$$\begin{aligned} \mathbf{H} &= \begin{bmatrix} \frac{x_1-X}{\rho_{r,1}} & \frac{y_1-Y}{\rho_{r,1}} & \frac{z_1-Z}{\rho_{r,1}} & 1 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{x_4-X}{\rho_{r,4}} & \frac{y_4-Y}{\rho_{r,4}} & \frac{z_4-Z}{\rho_{r,4}} & 1 \end{bmatrix} \\ \mathbf{A} &= (\mathbf{H}^T\mathbf{H})^{-1} \end{aligned} \tag{B.11}$$

We can now find:

$$\begin{aligned} \text{GDOP} &= \sqrt{a_{1,1} + a_{2,2} + a_{3,3} + a_{4,4}} \\ \text{PDOP} &= \sqrt{a_{1,1} + a_{2,2} + a_{3,3}} \end{aligned} \tag{B.12}$$

where $a_{i,j}$ indicates entries of matrix $\mathbf{A}$. For HDOP and VDOP we will need to perform coordinate transformations before calculating these matrices.

Lower DOP values indicate a more precise solution. Generally, DOP values will be lower when more GPS satellites are used. Table B.2 shows the quality associated with PDOP values.

Table B.2 – PDOP value indications [73].

PDOP range	<1	1-2	2-5	5-10	10-20	>20
Quality	Ideal case	Excellent	Good	Moderate	Fair	Poor

B.6 LEO Environment and Dynamics

The extreme velocities of LEO satellites are a major stumbling block in the development of their GPS receivers [9]. This is because of the large Doppler shift and shiftrates caused by these extreme velocities. Terrestrial receivers are built to accommodate carrier Doppler shifts of ±5 kHz, but a receiver on an LEO satellite can experience Doppler shifts of up to ±60 kHz [9]. Tracking loops in receivers typically have a 500 Hz bandwidth, and thus multiple frequency bins (each with a 500 Hz bandwidth) exist. Each bin also needs to search for a correct PRN code phase. This search period is called the dwell time. Predicting the Doppler shifts (using estimates of user and GPS satellites' position and velocity) reduces this dwell time. These bins need to be extended to be able to reach these extreme Doppler shifts. The acquisition process is illustrated in Figure B.5.

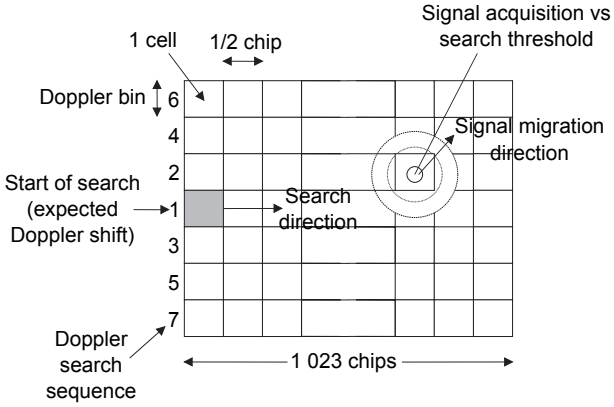


Figure B.5 – Illustration of search for correct chip offset and Doppler shift bin, obtained from [70].

A 18.3 km and 512 m/s limit on altitude and velocity is enforced by the International Traffic in Arms Regulations (ITAR). Official GPS module manufacturers thus include thresholds after which the receiver stops calculating position and velocity from the measurements. It is obvious that a LEO satellite's dynamics will surpass both of these regulations, and thus these limits must be removed from the GPS module's firmware before it can be used in space [9].

As mentioned before, the compensation for ionospheric delay should be removed from the firmware of a receiver intended to be spaceborne [9].

The Earth shadows terrestrial receivers from GPS satellites elevated below the horizon. Terrestrial receivers usually have an elevation mask of 5°, which means GPS satellites with predicted elevations lower than 5° are not used. This avoids the receiver using satellites which signal has been adversely affected by the ionosphere. However, as LEO satellites are far above the ionosphere, this elevation mask can be lowered to as much as -25°, depending on receiver antenna design [9].

B.7 TTFF

The time to first fix (TTFF) is the time from GPS turn-on to first navigation solution. This depends on the initial estimation and receiver characteristics. Three types of GPS start-up are defined: cold, warm and hot starts.

Cold Start

In this case no almanac or ephemeris data are available. Most of the time consumption goes into searching for satellites which are not visible to the receiver. The receiver has to search all the PRN codes and possible Doppler shifts for all satellites until a lock is found on one or more GPS satellites. When a lock is achieved, the almanac and ephemeris, taking 12.5 minutes and 30 s respectively, can be downloaded. A navigation solution will be found when either four or more satellites have been be locked onto, or the receiver waited the 12.5 minutes for the almanac, upon which the other satellites would be easy to track down. A cold start can typically take more than 15 minutes on older receivers [69].

Warm Start

If almanac data is available (valid for a few weeks), the Doppler offsets can be predicted with the last known position fix and current time. Typically, position, velocity and time should be known to within 100 km, 25 m/s and 20 s, respectively (this applies to terrestrial applications, and could be more stringent for the extreme dynamics of satellites). Searching for non-visible satellites is also avoided. Ephemeris data, essential for accurate solutions, can be downloaded within 30 s. TTFF for a warm start is typically between 2 and 5.5 minutes [69].

Hot Start

When the ephemeris (valid for 4 hours) and almanac are both available, a hot start can be made. The time, position and velocity should also be known to the meet requirements of the warm start. A navigation solution can be made as soon as a code and carrier lock is achieved. The TTFF of a hot start is normally in the order of 10 seconds [69].

Considering that our GPS module will typically not be put to sleep for more than 105 minutes (period corresponding to a 1000 km orbit), and the propagator will have relatively good estimates of our position and velocity, we can expect that we will mostly be doing hot starts. A 1-2 minute range for TTFF is a very safe approximation.

B.8 GPS Accuracy

A vast variety of GPS receiver technologies exist, and this makes it difficult to generalize the concept of GPS receiver accuracy. The fact that GPS error is not truly Gaussian and is time varying also complicates this issue. However, we will assume that regarding it as Gaussian is acceptable. Montenbruck [3] states that position errors, compared to the magnitude of their measurements, are typically 1 order more accurate than their velocity counterparts.

A number of measures for accuracy exist. The user-equivalent range error (UERE) is a budget of the expected error during the calculation of pseudorange. User position solution errors are usually split into horizontal and vertical components. The user position solutions are often given in 95% certainty values, which corresponds to a 2σ certainty in Gaussian noise terms.

Terrestrial

Since SA has been deactivated, the accuracy of the civil SPS service has vastly improved. Grewal [62] estimates a UERE budget of 5.6 m, 20.1 m and 36.1 m for navigation with P-code, C/A without SA and C/A with SA, respectively. NAVSTAR [69] budgets 13.0 m and $15.7 - 23.1$ m for P-code and C/A-code navigation, respectively.

According to Kaplan [70], SA typically degraded horizontal and vertical position estimates to the order of 50 m and 75 m (at a 95% probability level), where NAVSTAR [69] puts the horizontal accuracy at 100 m (95%). Upon the discontinuation of SA, Kaplan states that the SPS service is specified to be within 13 m horizontal, 22 m vertical and 40 ns in clock time (95%). Upon measurement, though, they achieved horizontal and vertical accuracies of 7.1 m and 11.4 m (95%), respectively.

According to NAVSTAR [69], PPS users can typically achieve 0.2 m/s (95%) per axis, and when SA is switched off, SPS users can achieve the same.

SumbandilaSat SGPS Receiver

Nortier [9] developed the SGPS receiver for SumbandilaSat. Simulator results (with deliberate errors) indicated the following expected accuracies for a spaceborne GPS receiver:

Table B.3 – Expected noise on SGPS receiver according to Nortier [9].

	σ_{Radial}	$\sigma_{\text{Along-track}}$	$\sigma_{\text{Cross-track}}$	σ	σ^2
Position	8.9747 m	3.0559 m	3.3621 m	10.06 m	101.2 m^2
Velocity	0.46739 m/s	0.16817 m/s	0.15502 m/s	0.5203 m/s	0.2708 m^2/s^2

B.9 Commercial Off-the-Shelf (COTS) GPS Receivers

Most commercial off-the-shelf (COTS) GPS receiver modules are not rated for the space environment. To qualify a GPS module for space use, the manufacturer has to include mitigation efforts to handle radiation, as well as remove the 18.3 km and 512 m/s navigation solution limitations set by ITAR (for which the GPS module manufacturers charge quite a hefty fee). A few GPS modules suitable for CubeSat missions are listed in Table B.4. As the ESL has experience with the NovAtel range of receivers, it is likely that the work of this thesis will be used in conjunction with their products.

Table B.4 – Examples of GPS modules on the market. [74; 75; 76; 77; 78; 79; 80]

Product code Manufacturer	Chan	Freq	Pos [m]	Vel [m/s]	Time [ns]	Power [W]	Dimensions Mass [mm] [g]	TTFF (hot, warm, cold) [s]
SGR-05U[i] SSTL	12	L_1	20[a,f]	0.25[a,f]	1000[a]	0.8	70 × 45 × 10 40	-, 90, 180
Phoenix-S[i] DLR	12	L_1	10[f]	0.1[f]	500	0.85	70 × 47 × 15 20	-, 60, 900
GPS12-V1-S[i] SpaceQuest	14	L_1	10[f]	0.03[b,f]	20	1.2	100 × 70 × 25 160	30, 45, 90
AsteRx-m[i] Septentrio	132	L_1+L_2	2.3[c,g]	0.015[c,g]	10[c]	0.6	70 × 48 × ±10 40	1.2, 20, 45
OEM615[i] NovAtel	120	L_1+L_2[d]	1.5[b,i,g]	0.03[b,g]	20[b]	1	71 × 46 × 11 24	35, -, 50
Copernicus II[j] Trimble	12	-	9.4[e,g]	0.06[g]	350	0.13	19 × 19 × 2.6 2	8, 35, 38
IT03[j] Fastrax	12	-	2.7[a,g]	0.1[g]	20[b]	0.1	22 × 24 × 2.6 2.7	4, 34, 35

[a] 95 % [b] RMS [c] 1σ [d] GPS+GLONASS [e] 90 % [f] LEO Spec [g] Terrestrial Spec
[h] Horizontal [i] PCB [j] IC

Appendix C

Mathematical concepts

C.1 Rotating Coordinates

This section is loosely based on the work of Thompson [26]. We first define:

$$\omega_e^i = \begin{bmatrix} 0 \\ 0 \\ \omega_e \end{bmatrix} \qquad \Omega_e^i = \begin{bmatrix} 0 & -\omega_e & 0 \\ \omega_e & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad C_e^i = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{C.1}$$

where ω_e^i is the three dimensional angular rate vector of ECEF relative to ECI frame, Ω_e^i is a 3×3 skew symmetrical matrix of three dimensional angular rate vectors of ECEF relative to ECI, C_e^i is the DCM transformation for position vectors from ECEF to ECI, and θ is the z rotation angle between the ECEF and ECI frames.

Note that

$$\begin{aligned} \dot{C}_e^i &= C_e^i \Omega_e^i \\ \ddot{C}_e^i &= C_e^i \dot{\Omega}_e^i + \dot{C}_e^i \Omega_e^i \\ &= C_e^i \dot{\Omega}_e^i + C_e^i \Omega_e^i \Omega_e^i \\ &= C_e^i \Omega_e^i \Omega_e^i \end{aligned} \tag{C.2}$$

$\dot{\Omega}_e^i$ is zero because we assume that Earth is rotating at a constant angular velocity.

We know that

$$\mathbf{r}_i = C_e^i \mathbf{r}_e \tag{C.3}$$

and by differentiating we obtain

$$\begin{aligned} \dot{\mathbf{r}}_i &= C_e^i \dot{\mathbf{r}}_e + \dot{C}_e^i \mathbf{r}_e \\ &= C_e^i \dot{\mathbf{r}}_e + C_e^i \Omega_e^i \mathbf{r}_e \\ &= C_e^i (\dot{\mathbf{r}}_e + \Omega_e^i \mathbf{r}_e) \\ &= C_e^i \begin{bmatrix} \dot{x}_{ECEF} - \omega_e y_{ECEF} \\ \dot{y}_{ECEF} + \omega_e x_{ECEF} \\ \dot{z}_{ECEF} \end{bmatrix} \end{aligned} \tag{C.4}$$

$$\begin{aligned}\ddot{\mathbf{r}}_i &= C_e^i\ddot{\mathbf{r}}_e + \dot{C}_e^i\dot{\mathbf{r}}_e + \dot{C}_e^i\dot{\mathbf{r}}_e + \ddot{C}_e^i\mathbf{r}_e \\ &= C_e^i(\ddot{\mathbf{r}}_e + 2\Omega_e^i\dot{\mathbf{r}}_e + \Omega_e^i\Omega_e^i\mathbf{r}_e) \\ &= C_e^i\begin{bmatrix}\ddot{x}_{ECEF} - 2\omega_e\dot{y}_{ECEF} - \omega_e^2 x_{ECEF} \\ \ddot{y}_{ECEF} + 2\omega_e\dot{x}_{ECEF} - \omega_e^2 y_{ECEF} \\ \ddot{z}_{ECEF}\end{bmatrix}\end{aligned} \tag{C.5}$$

C.2 6×6 Matrix Inversion

The EKF implemented for this paper requires the inversion of the matrix $(\mathbf{P}_k^- + \mathbf{R}_k)$, which is a 6×6 matrix. This task is very inefficient when done with ordinary methods. We propose the use of blockwise inversion (described in [81]), as an efficient alternative. This technique divides the problem up into two 3×3 inversion problems (whose analytical solutions are efficient) along with some normal matrix algebra, as shown in Equation C.6. Algorithm C.1 is used to perform the calculations.

$$\begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D}\end{bmatrix}^{-1} = \begin{bmatrix}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & \mathbf{D}^{-1}+\mathbf{D}^{-1}\mathbf{C}(\mathbf{A}-\mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1}\end{bmatrix} \tag{C.6}$$

Algorithm C.1 Blockwise inversion of a 6×6 matrix $\mathbf{M}$

Let $\mathbf{M} = \begin{bmatrix}\mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D}\end{bmatrix}$
$\mathbf{D}^{-1} = \text{inv}(\mathbf{D})$
$\mathbf{F} = \mathbf{B}\mathbf{D}^{-1}$
$\mathbf{G} = \mathbf{D}^{-1}\mathbf{C}$
$\mathbf{H} = \mathbf{F}\mathbf{C}$
$\mathbf{E} = \mathbf{A} - \mathbf{H}$
$\mathbf{E}^{-1} = \text{inv}(\mathbf{E})$
for i **do**=0 to 2
  for j **do**=0 to 2
    $\mathbf{M}_{i,j}^{-1} = \mathbf{E}_{i,j}^{-1}$
    $\mathbf{M}_{i,j+3}^{-1} = \mathbf{M}_{i+3,j} = \mathbf{M}_{i+3,j+3} = 0$
    for k **do**=0 to 2
      $\mathbf{M}_{i,j+3}^{-1} = \mathbf{M}_{i,j+3}^{-1} - \mathbf{E}_{i,k}^{-1}\mathbf{F}_{k,j}$
      $\mathbf{M}_{i+3,j}^{-1} = \mathbf{M}_{i+3,j}^{-1} - \mathbf{G}_{i,k}\mathbf{E}_{k,j}^{-1}$
    end for
  end for
end for
for i **do**=0 to 2
  for j **do**=0 to 2
    for k **do**=0 to 2
      $\mathbf{M}_{i+3,j+3}^{-1} = \mathbf{M}_{i+3,j+3}^{-1} - \mathbf{G}_{i,k}\mathbf{M}_{k,j+3}^{-1}$
    end for
    $\mathbf{M}_{i+3,j+3}^{-1} = \mathbf{M}_{i+3,j+3}^{-1} + \mathbf{D}_{i,j}^{-1}$
  end for
end for

The basic analytical algorithm for calculating the inverse of a 3×3 matrix is:

Algorithm C.2 Inverting a 3×3 matrix $\mathbf{A}$

Let $\mathbf{A} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & k \end{bmatrix}$

$A = (ek - fh)$

$B = (fg - dk)$

$C = (dh - eg)$

$\det(\mathbf{A}) = aA + bB + cC$

Then $\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{bmatrix} A & (ch - bk) & (bf - ce) \\ B & (ak - cg) & (cd - af) \\ C & (gb - ah) & (ae - bd) \end{bmatrix}$

C.3 IIR Filter

An infinite impulse response (IIR) filter is a very simple recursive digital filter that requires minimal computational efforts. The general IIR filter can be described by [82; 83]:

$$y[n] = \sum_{l=1}^{N} a_l y[n-l] + \sum_{k=0}^{M} b_k x[n-k] \tag{C.7}$$

For a very simple IIR filter let's use $N = 1$ and $M = 0$. The impulse response of this simple IIR can easily found as [83]:

$$h[n] = b_0 (a_1)^n u[n] \tag{C.8}$$

where $u[n]$ simply indicates the causality of the filter. It is clear that we want $a_1 < 1$ to ensure a stable response.

We must ensure that the step response tends to 1 (to follow the signal). Assuming that $a_1 < 1$ (actually holds for any $a_1 \neq 1$), the step response is:

$$y[n] = b_0 \frac{1 - a_1^{n+1}}{1 - a_1} u[n] \tag{C.9}$$

Using the step response limit, we find the condition to ensure this:

$$\begin{aligned} \lim_{n \to \infty} y[n] &= \lim_{n \to \infty} b_0 \frac{1 - a_1^{n+1}}{1 - a_1} u[n] = b_0 \frac{1}{1 - a_1} = 1 \\ b_0 &= 1 - a_1 \\ a_1 + b_0 &= 1 \end{aligned} \tag{C.10}$$

We introduce α and define our first order IIR filter function as:

$$y[n] = \alpha x[n] + (1 - \alpha) y[n-1] \tag{C.11}$$

In this form, increasing α yields a faster time response. Choosing $\alpha > 1$ causes an overshoot, and $\alpha > 2$ or $\alpha < 0$ produces unstable systems.

Appendix D

Data Tables

D.1 SMAD Exponential Atmospheric Density Model

Table D.1 – SMAD exponential atmospheric density model [14].

Altitude	Base Alt	Nominal Density			Scale Height		
	h_0	ρ_0			H		
km	km	kg/m^3			km		
		Min	Ave	Max	Min	Ave	Max
100-150	100	5.71×10^{-7}	5.69×10^{-7}	5.67×10^{-7}	5.8	5.8	5.8
150-175	150	1.9×10^{-9}	2.02×10^{-9}	2.21×10^{-9}	21.7	24.6	27.9
175-200	175	6.42×10^{-10}	7.66×10^{-10}	9.21×10^{-10}	25.5	29.9	34.8
200-225	200	2.18×10^{-10}	2.9×10^{-10}	3.84×10^{-10}	29.4	35.3	41.6
225-250	225	9.64×10^{-11}	1.46×10^{-10}	2.12×10^{-10}	31.7	38.4	45.5
250-275	250	4.27×10^{-11}	7.3×10^{-11}	1.17×10^{-10}	34.1	41.4	49.4
275-300	275	2.14×10^{-11}	4.1×10^{-11}	7.17×10^{-11}	35.9	43.6	51.9
300-325	300	1.07×10^{-11}	2.3×10^{-11}	4.39×10^{-11}	37.7	45.8	54.5
325-350	325	5.83×10^{-12}	1.38×10^{-11}	2.85×10^{-11}	39.1	47.4	56.4
350-375	350	3.17×10^{-12}	8.33×10^{-12}	1.85×10^{-11}	40.5	49	58.3
375-400	375	1.81×10^{-12}	5.24×10^{-12}	1.25×10^{-11}	42	50.3	59.7
400-450	400	1.04×10^{-12}	3.29×10^{-12}	8.43×10^{-12}	43.5	51.7	61.1
450-500	450	3.68×10^{-13}	1.39×10^{-12}	4.05×10^{-12}	48.1	54.2	63.5
500-550	500	1.4×10^{-13}	6.15×10^{-13}	2.03×10^{-12}	56.6	57.3	65.6
550-600	550	5.76×10^{-14}	2.84×10^{-13}	1.05×10^{-12}	71.1	62	67.9
600-650	600	2.61×10^{-14}	1.37×10^{-13}	5.63×10^{-13}	92.1	69.3	70.7
650-700	650	1.32×10^{-14}	6.87×10^{-14}	3.08×10^{-13}	116.8	80.5	74.7
700-750	700	7.55×10^{-15}	3.63×10^{-14}	1.73×10^{-13}	141.3	95.9	80.5
750-800	750	4.81×10^{-15}	2.02×10^{-14}	9.95×10^{-14}	163.7	114.6	88.6
800-850	800	3.34×10^{-15}	1.21×10^{-14}	5.88×10^{-14}	183.87	134.8	99.5
850-900	850	2.47×10^{-15}	7.69×10^{-15}	3.57×10^{-14}	202.67	154.77	113.09
900-950	900	1.9×10^{-15}	5.24×10^{-15}	2.25×10^{-14}	220.77	173.43	128.76
950-1000	950	1.5×10^{-15}	3.78×10^{-15}	1.46×10^{-14}	239.66	190.33	145.55

D.2 JGM-3 Earth Gravity Model

Table D.2 – JGM-3 Earth gravity model [19].

n	m	$C_{n,m}$	$S_{n,m}$
2	0	$-1.082\,626\,690\,6 \times 10^{-3}$	-
2	1	$-2.414\,000 \times 10^{-10}$	$1.543\,100 \times 10^{-9}$
2	2	$1.574\,536 \times 10^{-6}$	$-9.038\,681 \times 10^{-7}$
3	0	$2.532\,435 \times 10^{-6}$	-
3	1	$2.192\,799 \times 10^{-6}$	$2.680\,119 \times 10^{-7}$
3	2	$3.090\,160 \times 10^{-7}$	$-2.114\,024 \times 10^{-7}$
3	3	$1.005\,589 \times 10^{-7}$	$1.972\,013 \times 10^{-7}$
4	0	$1.619\,331 \times 10^{-6}$	-
4	1	$-5.087\,253 \times 10^{-7}$	$-4.494\,599 \times 10^{-7}$
4	2	$7.841\,223 \times 10^{-8}$	$1.481\,555 \times 10^{-7}$
4	3	$5.921\,574 \times 10^{-8}$	$-1.201\,129 \times 10^{-8}$
4	4	$-3.982\,396 \times 10^{-9}$	$6.525\,606 \times 10^{-9}$
5	0	$2.277\,161 \times 10^{-7}$	-
5	1	$-5.371\,651 \times 10^{-8}$	$-8.066\,346 \times 10^{-8}$
5	2	$1.055\,905 \times 10^{-7}$	$-5.232\,672 \times 10^{-8}$
5	3	$-1.492\,615 \times 10^{-8}$	$-7.100\,877 \times 10^{-9}$
5	4	$-2.297\,912 \times 10^{-9}$	$3.873\,005 \times 10^{-10}$
5	5	$4.304\,768 \times 10^{-10}$	$-1.648\,204 \times 10^{-9}$
6	0	$-5.396\,485 \times 10^{-7}$	-
6	1	$-5.987\,798 \times 10^{-8}$	$2.116\,466 \times 10^{-8}$
6	2	$6.012\,099 \times 10^{-9}$	$-4.650\,395 \times 10^{-8}$
6	3	$1.182\,266 \times 10^{-9}$	$1.843\,134 \times 10^{-10}$
6	4	$-3.264\,139 \times 10^{-10}$	$-1.784\,491 \times 10^{-9}$
6	5	$-2.155\,771 \times 10^{-10}$	$-4.329\,182 \times 10^{-10}$
6	6	$2.213\,693 \times 10^{-12}$	$-5.527\,712 \times 10^{-11}$
7	0	$3.513\,684 \times 10^{-7}$	-
7	1	$2.051\,487 \times 10^{-7}$	$6.936\,990 \times 10^{-8}$
7	2	$3.284\,491 \times 10^{-8}$	$9.282\,314 \times 10^{-9}$
7	3	$3.528\,541 \times 10^{-9}$	$-3.061\,150 \times 10^{-9}$
7	4	$-5.851\,195 \times 10^{-10}$	$-2.636\,182 \times 10^{-10}$
7	5	$5.818\,485 \times 10^{-13}$	$6.397\,253 \times 10^{-12}$
7	6	$-2.490\,718 \times 10^{-11}$	$1.053\,488 \times 10^{-11}$
7	7	$2.559\,078 \times 10^{-14}$	$4.475\,983 \times 10^{-13}$
8	0	$2.025\,187 \times 10^{-7}$	-
8	1	$1.603\,459 \times 10^{-8}$	$4.019\,978 \times 10^{-8}$
8	2	$6.576\,542 \times 10^{-9}$	$5.381\,316 \times 10^{-9}$
8	3	$-1.946\,358 \times 10^{-10}$	$-8.723\,520 \times 10^{-10}$
8	4	$-3.189\,358 \times 10^{-10}$	$9.117\,736 \times 10^{-11}$
8	5	$-4.615\,173 \times 10^{-12}$	$1.612\,521 \times 10^{-11}$
8	6	$-1.839\,364 \times 10^{-12}$	$8.627\,743 \times 10^{-12}$
8	7	$3.429\,762 \times 10^{-13}$	$3.814\,766 \times 10^{-13}$
8	8	$-1.580\,332 \times 10^{-13}$	$1.535\,338 \times 10^{-13}$
9	0	$1.193\,687 \times 10^{-7}$	-
9	1	$9.241\,927 \times 10^{-8}$	$1.423\,657 \times 10^{-8}$
9	2	$1.566\,874 \times 10^{-9}$	$-2.228\,679 \times 10^{-9}$
9	3	$-1.217\,275 \times 10^{-9}$	$-5.633\,922 \times 10^{-10}$
9	4	$-7.018\,561 \times 10^{-12}$	$1.717\,309 \times 10^{-11}$
9	5	$-1.669\,737 \times 10^{-12}$	$-5.550\,919 \times 10^{-12}$
9	6	$8.296\,725 \times 10^{-13}$	$2.940\,313 \times 10^{-12}$
9	7	$-2.251\,973 \times 10^{-13}$	$-1.846\,792 \times 10^{-13}$
9	8	$6.144\,394 \times 10^{-14}$	$-9.856\,184 \times 10^{-16}$
9	9	$-3.676\,763 \times 10^{-15}$	$7.441\,039 \times 10^{-15}$
10	0	$2.480\,569 \times 10^{-7}$	-
10	1	$5.175\,579 \times 10^{-8}$	$-8.128\,915 \times 10^{-8}$
10	2	$-5.562\,846 \times 10^{-9}$	$-3.057\,129 \times 10^{-9}$
10	3	$-4.195\,999 \times 10^{-11}$	$-8.989\,333 \times 10^{-10}$
10	4	$-4.967\,025 \times 10^{-11}$	$-4.622\,483 \times 10^{-11}$
10	5	$-3.074\,283 \times 10^{-12}$	$-3.122\,269 \times 10^{-12}$
10	6	$-2.597\,232 \times 10^{-13}$	$-5.515\,591 \times 10^{-13}$
10	7	$6.909\,154 \times 10^{-15}$	$-2.650\,681 \times 10^{-15}$
10	8	$4.635\,314 \times 10^{-15}$	$-1.052\,843 \times 10^{-14}$
10	9	$2.330\,148 \times 10^{-15}$	$-7.011\,948 \times 10^{-16}$
10	10	$4.170\,802 \times 10^{-16}$	$-9.892\,610 \times 10^{-17}$
11	0	$-2.405\,652 \times 10^{-7}$	-
11	1	$9.508\,428 \times 10^{-9}$	$-1.646\,547 \times 10^{-8}$
11	2	$9.542\,030 \times 10^{-10}$	$-5.097\,360 \times 10^{-9}$
11	3	$-1.409\,608 \times 10^{-10}$	$-6.863\,521 \times 10^{-10}$
11	4	$-1.685\,257 \times 10^{-11}$	$-2.677\,798 \times 10^{-11}$
11	5	$1.489\,441 \times 10^{-12}$	$1.982\,505 \times 10^{-12}$
11	6	$-5.754\,671 \times 10^{-15}$	$1.346\,234 \times 10^{-13}$
11	7	$1.954\,262 \times 10^{-15}$	$-3.728\,037 \times 10^{-14}$

Table D.2 – JGM-3 Earth gravity model (cont.) [19].

n	m	$C_{n,m}$	$S_{n,m}$
11	8	$-2.924\,949 \times 10^{-16}$	$1.170\,448 \times 10^{-15}$
11	9	$-1.934\,320 \times 10^{-16}$	$2.585\,245 \times 10^{-16}$
11	10	$-4.946\,396 \times 10^{-17}$	$-1.736\,649 \times 10^{-17}$
11	11	$9.351\,705 \times 10^{-18}$	$-1.407\,856 \times 10^{-17}$
12	0	$1.819\,117 \times 10^{-7}$	-
12	1	$-3.068\,001 \times 10^{-8}$	$-2.378\,448 \times 10^{-8}$
12	2	$6.380\,397 \times 10^{-10}$	$1.416\,422 \times 10^{-9}$
12	3	$1.451\,918 \times 10^{-10}$	$9.154\,575 \times 10^{-11}$
12	4	$-2.123\,815 \times 10^{-11}$	$9.170\,517 \times 10^{-13}$
12	5	$8.279\,902 \times 10^{-13}$	$2.033\,249 \times 10^{-13}$
12	6	$7.883\,092 \times 10^{-15}$	$9.335\,408 \times 10^{-14}$
12	7	$-4.131\,557 \times 10^{-15}$	$7.899\,913 \times 10^{-15}$
12	8	$-5.708\,254 \times 10^{-16}$	$3.701\,522 \times 10^{-16}$
12	9	$1.012\,728 \times 10^{-16}$	$6.136\,644 \times 10^{-17}$
12	10	$-1.840\,173 \times 10^{-18}$	$9.242\,424 \times 10^{-18}$
12	11	$4.978\,700 \times 10^{-19}$	$-2.790\,078 \times 10^{-19}$
12	12	$-2.108\,949 \times 10^{-20}$	$-9.838\,299 \times 10^{-20}$
13	0	$2.075\,677 \times 10^{-7}$	-
13	1	$-2.885\,131 \times 10^{-8}$	$2.172\,109 \times 10^{-8}$
13	2	$2.275\,183 \times 10^{-9}$	$-2.545\,587 \times 10^{-9}$
13	3	$-6.676\,767 \times 10^{-11}$	$3.005\,522 \times 10^{-10}$
13	4	$-3.452\,536 \times 10^{-13}$	$-2.960\,682 \times 10^{-12}$
13	5	$1.074\,251 \times 10^{-12}$	$1.214\,266 \times 10^{-12}$
13	6	$-5.281\,862 \times 10^{-14}$	$-9.061\,871 \times 10^{-15}$
13	7	$3.421\,269 \times 10^{-16}$	$-9.747\,983 \times 10^{-16}$
13	8	$-1.113\,494 \times 10^{-16}$	$-1.095\,673 \times 10^{-16}$
13	9	$2.658\,019 \times 10^{-17}$	$4.870\,630 \times 10^{-17}$
13	10	$4.577\,888 \times 10^{-18}$	$-4.153\,238 \times 10^{-18}$
13	11	$-5.902\,637 \times 10^{-19}$	$-6.376\,262 \times 10^{-20}$
13	12	$-5.860\,602 \times 10^{-20}$	$1.643\,922 \times 10^{-19}$
13	13	$-2.239\,852 \times 10^{-20}$	$2.503\,221 \times 10^{-20}$
14	0	$-1.174\,174 \times 10^{-7}$	-
14	1	$-9.997\,709 \times 10^{-9}$	$1.443\,750 \times 10^{-8}$
14	2	$-1.347\,496 \times 10^{-9}$	$-1.089\,216 \times 10^{-10}$
14	3	$9.391\,107 \times 10^{-11}$	$5.182\,512 \times 10^{-11}$
14	4	$3.104\,170 \times 10^{-13}$	$-3.750\,977 \times 10^{-12}$
14	5	$3.932\,888 \times 10^{-13}$	$-2.217\,440 \times 10^{-13}$
14	6	$-1.902\,110 \times 10^{-14}$	$2.365\,712 \times 10^{-15}$
14	7	$2.787\,457 \times 10^{-15}$	$-3.193\,839 \times 10^{-16}$
14	8	$-2.125\,248 \times 10^{-16}$	$-9.074\,975 \times 10^{-17}$
14	9	$1.679\,922 \times 10^{-17}$	$1.489\,060 \times 10^{-17}$
14	10	$1.839\,624 \times 10^{-18}$	$-6.937\,464 \times 10^{-20}$
14	11	$7.273\,780 \times 10^{-20}$	$-1.849\,098 \times 10^{-19}$
14	12	$4.561\,174 \times 10^{-21}$	$-1.658\,377 \times 10^{-20}$
14	13	$2.347\,630 \times 10^{-21}$	$3.298\,844 \times 10^{-21}$
14	14	$-7.142\,240 \times 10^{-22}$	$-6.914\,977 \times 10^{-23}$
15	0	$1.762\,727 \times 10^{-8}$	-
15	1	$6.108\,862 \times 10^{-9}$	$4.154\,186 \times 10^{-9}$
15	2	$-7.164\,511 \times 10^{-10}$	$-1.045\,474 \times 10^{-9}$
15	3	$1.128\,627 \times 10^{-10}$	$3.265\,044 \times 10^{-11}$
15	4	$-6.013\,879 \times 10^{-12}$	$1.116\,419 \times 10^{-12}$
15	5	$1.293\,500 \times 10^{-13}$	$8.637\,823 \times 10^{-14}$
15	6	$2.220\,625 \times 10^{-14}$	$-2.505\,252 \times 10^{-14}$
15	7	$2.825\,477 \times 10^{-15}$	$2.856\,094 \times 10^{-16}$
15	8	$-1.112\,172 \times 10^{-16}$	$7.742\,870 \times 10^{-17}$
15	9	$3.494\,173 \times 10^{-18}$	$1.015\,963 \times 10^{-17}$
15	10	$2.258\,283 \times 10^{-19}$	$3.275\,583 \times 10^{-19}$
15	11	$-1.828\,153 \times 10^{-21}$	$3.595\,115 \times 10^{-20}$
15	12	$-6.049\,406 \times 10^{-21}$	$2.905\,537 \times 10^{-21}$
15	13	$-5.705\,023 \times 10^{-22}$	$-8.660\,491 \times 10^{-23}$
15	14	$1.404\,654 \times 10^{-23}$	$-6.472\,496 \times 10^{-23}$
15	15	$-9.295\,856 \times 10^{-24}$	$-2.274\,403 \times 10^{-24}$
16	0	$-3.119\,431 \times 10^{-8}$	-
16	1	$1.356\,279 \times 10^{-8}$	$1.660\,440 \times 10^{-8}$
16	2	$-6.713\,707 \times 10^{-10}$	$7.856\,272 \times 10^{-10}$
16	3	$-6.451\,812 \times 10^{-11}$	$-4.271\,981 \times 10^{-11}$
16	4	$4.698\,674 \times 10^{-12}$	$5.250\,141 \times 10^{-12}$
16	5	$-9.690\,791 \times 10^{-14}$	$-1.205\,563 \times 10^{-14}$
16	6	$6.610\,666 \times 10^{-15}$	$-1.590\,014 \times 10^{-14}$
16	7	$-2.378\,057 \times 10^{-16}$	$-2.590\,259 \times 10^{-16}$
16	8	$-4.460\,480 \times 10^{-17}$	$1.086\,771 \times 10^{-17}$
16	9	$-3.335\,458 \times 10^{-18}$	$-5.700\,076 \times 10^{-18}$
16	10	$-1.316\,568 \times 10^{-19}$	$1.309\,613 \times 10^{-19}$
16	11	$1.643\,081 \times 10^{-20}$	$-2.537\,013 \times 10^{-21}$
16	12	$1.419\,788 \times 10^{-21}$	$4.983\,891 \times 10^{-22}$
16	13	$9.260\,415 \times 10^{-23}$	$6.651\,727 \times 10^{-24}$
16	14	$-1.349\,210 \times 10^{-23}$	$-2.741\,331 \times 10^{-23}$
16	15	$-1.295\,522 \times 10^{-24}$	$-2.929\,523 \times 10^{-24}$
16	16	$-5.943\,715 \times 10^{-25}$	$5.687\,404 \times 10^{-26}$
17	0	$1.071\,306 \times 10^{-7}$	-

Table D.2 – JGM-3 Earth gravity model (cont.) [19].

n	m	$C_{n,m}$	$S_{n,m}$
17	1	$-1.262\,144 \times 10^{-8}$	$-1.427\,822 \times 10^{-8}$
17	2	$-4.767\,232 \times 10^{-10}$	$2.522\,818 \times 10^{-10}$
17	3	$1.175\,560 \times 10^{-11}$	$1.297\,841 \times 10^{-11}$
17	4	$6.946\,241 \times 10^{-13}$	$2.159\,727 \times 10^{-12}$
17	5	$-9.316\,734 \times 10^{-14}$	$2.923\,804 \times 10^{-14}$
17	6	$-4.427\,290 \times 10^{-15}$	$-9.295\,650 \times 10^{-15}$
17	7	$4.858\,365 \times 10^{-16}$	$-1.190\,467 \times 10^{-16}$
17	8	$4.814\,810 \times 10^{-17}$	$4.812\,890 \times 10^{-18}$
17	9	$2.752\,709 \times 10^{-19}$	$-2.391\,386 \times 10^{-18}$
17	10	$-2.449\,926 \times 10^{-20}$	$1.026\,767 \times 10^{-19}$
17	11	$-6.393\,665 \times 10^{-21}$	$4.480\,853 \times 10^{-21}$
17	12	$8.842\,755 \times 10^{-22}$	$6.393\,876 \times 10^{-22}$
17	13	$4.178\,428 \times 10^{-23}$	$5.110\,031 \times 10^{-23}$
17	14	$-3.177\,778 \times 10^{-24}$	$2.570\,941 \times 10^{-24}$
17	15	$1.229\,863 \times 10^{-25}$	$1.242\,605 \times 10^{-25}$
17	16	$-8.535\,124 \times 10^{-26}$	$1.057\,368 \times 10^{-26}$
17	17	$-1.658\,684 \times 10^{-26}$	$-9.608\,699 \times 10^{-27}$
18	0	$4.421\,672 \times 10^{-8}$	-
18	1	$1.958\,333 \times 10^{-9}$	$-1.817\,656 \times 10^{-8}$
18	2	$3.236\,166 \times 10^{-10}$	$3.427\,413 \times 10^{-10}$
18	3	$-5.174\,199 \times 10^{-12}$	$-4.278\,803 \times 10^{-12}$
18	4	$4.022\,242 \times 10^{-12}$	$1.105\,860 \times 10^{-13}$
18	5	$3.088\,082 \times 10^{-14}$	$1.040\,715 \times 10^{-13}$
18	6	$3.197\,551 \times 10^{-15}$	$-3.743\,268 \times 10^{-15}$
18	7	$9.009\,281 \times 10^{-17}$	$8.666\,599 \times 10^{-17}$
18	8	$2.534\,982 \times 10^{-17}$	$2.015\,619 \times 10^{-18}$
18	9	$-9.526\,324 \times 10^{-19}$	$1.794\,927 \times 10^{-18}$
18	10	$1.741\,250 \times 10^{-20}$	$-1.437\,566 \times 10^{-20}$
18	11	$-1.569\,624 \times 10^{-21}$	$4.348\,241 \times 10^{-22}$
18	12	$-4.195\,542 \times 10^{-22}$	$-2.294\,907 \times 10^{-22}$
18	13	$-6.629\,972 \times 10^{-24}$	$-3.635\,064 \times 10^{-23}$
18	14	$-6.574\,751 \times 10^{-25}$	$-1.074\,458 \times 10^{-24}$
18	15	$-2.898\,577 \times 10^{-25}$	$-1.447\,976 \times 10^{-25}$
18	16	$7.555\,273 \times 10^{-27}$	$4.931\,703 \times 10^{-27}$
18	17	$3.046\,776 \times 10^{-28}$	$3.816\,913 \times 10^{-28}$
18	18	$3.696\,154 \times 10^{-29}$	$-1.524\,672 \times 10^{-28}$
19	0	$-2.197\,334 \times 10^{-8}$	-
19	1	$-3.156\,695 \times 10^{-9}$	$7.160\,542 \times 10^{-11}$
19	2	$7.325\,273 \times 10^{-10}$	$-1.008\,909 \times 10^{-10}$
19	3	$-1.192\,913 \times 10^{-11}$	$-1.190\,759 \times 10^{-12}$
19	4	$9.941\,288 \times 10^{-13}$	$-3.556\,436 \times 10^{-13}$
19	5	$3.991\,922 \times 10^{-14}$	$9.006\,136 \times 10^{-14}$
19	6	$-4.220\,405 \times 10^{-16}$	$3.176\,649 \times 10^{-15}$
19	7	$7.091\,584 \times 10^{-17}$	$-8.340\,023 \times 10^{-17}$
19	8	$1.660\,451 \times 10^{-17}$	$-5.594\,661 \times 10^{-18}$
19	9	$9.233\,532 \times 10^{-20}$	$1.965\,726 \times 10^{-19}$
19	10	$-5.971\,907 \times 10^{-20}$	$-1.268\,576 \times 10^{-20}$
19	11	$1.750\,987 \times 10^{-21}$	$1.197\,796 \times 10^{-21}$
19	12	$-2.066\,463 \times 10^{-23}$	$6.437\,043 \times 10^{-23}$
19	13	$-3.440\,194 \times 10^{-24}$	$-1.311\,958 \times 10^{-23}$
19	14	$-1.487\,095 \times 10^{-25}$	$-4.305\,386 \times 10^{-25}$
19	15	$-4.491\,878 \times 10^{-26}$	$-3.551\,991 \times 10^{-26}$
19	16	$-4.558\,801 \times 10^{-27}$	$-1.480\,665 \times 10^{-27}$
19	17	$5.960\,375 \times 10^{-28}$	$-3.102\,988 \times 10^{-28}$
19	18	$8.263\,952 \times 10^{-29}$	$-2.246\,909 \times 10^{-29}$
19	19	$-9.155\,723 \times 10^{-31}$	$1.845\,778 \times 10^{-30}$
20	0	$1.203\,146 \times 10^{-7}$	-
20	1	$3.688\,524 \times 10^{-9}$	$2.759\,192 \times 10^{-9}$
20	2	$4.328\,972 \times 10^{-10}$	$3.216\,826 \times 10^{-10}$
20	3	$-6.303\,973 \times 10^{-12}$	$3.778\,260 \times 10^{-11}$
20	4	$2.869\,668 \times 10^{-13}$	$-1.178\,441 \times 10^{-12}$
20	5	$-3.011\,115 \times 10^{-14}$	$-1.823\,414 \times 10^{-14}$
20	6	$1.539\,793 \times 10^{-15}$	$-5.637\,287 \times 10^{-17}$
20	7	$-1.390\,222 \times 10^{-16}$	$-8.899\,419 \times 10^{-19}$
20	8	$1.766\,707 \times 10^{-18}$	$1.459\,810 \times 10^{-18}$
20	9	$3.471\,731 \times 10^{-19}$	$-1.128\,428 \times 10^{-19}$
20	10	$-3.447\,438 \times 10^{-20}$	$-6.100\,911 \times 10^{-21}$
20	11	$8.760\,348 \times 10^{-22}$	$-1.138\,734 \times 10^{-21}$
20	12	$-2.271\,884 \times 10^{-23}$	$6.435\,154 \times 10^{-23}$
20	13	$5.960\,951 \times 10^{-24}$	$1.534\,228 \times 10^{-24}$
20	14	$1.682\,025 \times 10^{-25}$	$-2.046\,569 \times 10^{-25}$
20	15	$-2.520\,877 \times 10^{-26}$	$-7.473\,051 \times 10^{-28}$
20	16	$-8.774\,567 \times 10^{-28}$	$2.400\,401 \times 10^{-29}$
20	17	$2.651\,434 \times 10^{-29}$	$-8.192\,993 \times 10^{-29}$
20	18	$8.352\,807 \times 10^{-30}$	$-5.508\,346 \times 10^{-31}$
20	19	$-1.878\,413 \times 10^{-31}$	$6.948\,820 \times 10^{-31}$
20	20	$4.054\,696 \times 10^{-32}$	$-1.237\,749 \times 10^{-31}$

Bibliography

[1] Greene, M. and Zee, R.: Increasing the Accuracy of Orbital Position Information from NORAD SGP4 using Intermittent GPS Readings. *AIAA/USU Conference on Small Satellites*, 2009.

[2] Vetter, J.R.: Fifty Years of Orbit Determination: Development of Modern Astrodynamics Methods. *Johns Hopkins APL Technical Digest*, vol. 27, no. 3, p. 239, 2007.

[3] Montenbruck, O.: An Epoch State Filter for use with Analytical Orbit Models of Low Earth Satellites. *Aerospace Science and Technology*, vol. 4, no. 4, pp. 277–287, 2000.

[4] Mehrparvar, A. *et al.*: Cubesat design specification rev. 13. *The CubeSat Program, California Polytechnic State University*, vol. 1, 2009.

[5] Clyde Space SP-L-S1U-0002-CS Solar Panel Datasheet. Sep 2015.
Available at: `http://www.clyde-space.com/cubesat_shop/solar_panels/1u_solar_panels/50_1u-cubesat-side-solar-panel`

[6] NanoPower P110 Series Solar Panel Datasheet. Sep 2015.
Available at: `http://gomspace.com/documents/GS-DS-P110-1.0.pdf`

[7] Kahr, E., Skone, S. and O'Keefe, K.: Orbit Determination for the Canx-2 Nanosatellite using Intermittent GPS Data. *Proceedings of ION GNSS, Portland, OR*, 2010.

[8] Image of Earth in space. Sep 2015.
Available at: `https://0.s3.envato.com/files/114612207/Flight%20Earth%20In%20Space.jpg`

[9] Nortier, B.J.: *A Spaceborne GPS Receiver*. MSc., Stellenbosch University, Dec 2003.

[10] Kelso, T.: CelesTrak. `http://www.celestrak.com/columns/v04n03/`, March 2015.

[11] *CubeComputer Datasheet*, Nov 2015.
Available at: `http://www.cubespace.co.za/#!components/c49`

[12] Vallado, D.A.: *Fundamentals of Astrodynamics and Applications*. 2nd edn. Micocosm Press and Kluwer Academic Publishers, 2001.

[13] Montenbruck, O. and Gill, E.: *Satellite Orbits: Models, Methods and Applications*. Springer Science & Business Media, 2000.

[14] Wertz, J.R., Everett, D.F. and Puschell, J.J. (eds.): *Space Mission Engineering: The New SMAD*. 4th edn. Micocosm Press, 2011.

[15] Rapp, R.H.: Past and Future Developments in Geopotential Modelling. In: *Geodesy on the Move*, pp. 58–78. Springer, 1998.

[16] Tapley, B.D., Watkins, M., Ries, J., Davis, G., Eanes, R., Poole, S., Rim, H., Schutz, B., Shum, C., Nerem, R. *et al.*: The Joint Gravity Model 3. *Journal of Geophysical Research: Solid Earth (1978–2012)*, vol. 101, no. B12, pp. 28029–28049, 1996.

[17] Pavlis, N.K., Holmes, S.A., Kenyon, S.C. and Factor, J.K.: The Development and Evaluation of the Earth Gravitational Model 2008 (EGM2008). *Journal of Geophysical Research: Solid Earth (1978–2012)*, vol. 117, no. B4, 2012.

[18] Cunningham, L.E.: On the Computation of the Spherical Harmonic Terms Needed During the Numerical Integration of the Orbit Motion of an Artificial Satellite. *Celestial Mechanics*, vol. 2, no. 2, pp. 207–216, 1970.

[19] SphericalHarmonicGravity.cpp source code part of GPSTK, the GPS Toolkit. `http://www.gpstk.org/doxygen/SphericalHarmonicGravity_8cpp-source.html`, Jun 2012.

[20] Kuga, H.K. and Carrara, V.: Fortran- and C-codes for higher order and degree geopotential and derivatives computation. In: *Proceedings of XVI SBSR-Brazilian Symposium on Remote Sensing, Iguassu Falls, Brazil, April*, pp. 13–18. Apr 2013.

[21] About Atmospheric Modelweb Models at CCMC. `http://ccmc.gsfc.nasa.gov/modelweb/atmos/about_atmos.html`, Feb 2015.

[22] Doornbos, E.N.: *Thermospheric Density and Wind Determination from Satellite Dynamics*. Ph.D., Delft University of Technology, 2001.

[23] Tapping, K.: The 10.7 cm Solar Radio Flux (F10.7). *Space Weather*, vol. 11, no. 7, pp. 394–406, 2013.

[24] Bowman, B.R., Tobiska, W.K., Marcos, F.A., Huang, C.Y., Lin, C.S. and Burke, W.J.: A new Empirical Thermospheric Density Model JB2008 using new Solar and Geomagnetic Indices. In: *AIAA/AAS Astrodynamics Specialist Conference and Exhibit, Honolulu, Hawaii*. 2008.

[25] Dowd, D.L. and Tapley, B.: Density Models for the Upper Atmosphere. *Celestial mechanics*, vol. 20, no. 3, pp. 271–295, Jan 1979.

[26] Thompson, A.B.: Navigation Equations in the Earth Centered Earth Fixed Frame. `http://www.atacolorado.com/NavigationEquationsInTheEarthCenteredEarthFixedFrame.doc`, Nov 2014.

[27] Approximate Solar Coordinates. `http://aa.usno.navy.mil/faq/docs/SunApprox.php`, Nov 2012. Astronomical Applications Department, U.S. Naval Observatory.

[28] Cho, C.-H., Lee, B.-S., Lee, J.-S., Kim, J.-H. and Choi, K.-H.: NORAD TLE Type Orbit Determination of LEO Satellites using GPS Navigation Solutions. *Journal of Astronomy and Space Sciences*, vol. 19, pp. 197–206, 2002.

[29] Vallado, D.A., Crawford, P., Hujsak, R. and Kelso, T.: Revisiting Spacetrack Report# 3. *AIAA*, vol. 6753, p. 2006, 2006.

[30] Hoots, F.R., Jr., P.W.S. and Glover, R.A.: A History of Analytical Orbit Modelling in the United States Space Surveillance System. *Journal of Guidance, Control, and Dynamics*, Apr 2004.

[31] Kahr, E., Montenbruck, O. and O'Keefe, K.P.: Estimation and Analysis of Two-line Elements for Small Satellites. *Journal of Spacecraft and Rockets*, vol. 50, no. 2, pp. 433–439, 2013.

[32] Dismukes, K.: Definition of Two-line Element Set Coordinate System. `http://spaceflight.nasa.gov/realdata/sightings/SSapplications/Post/JavaSSOP/SSOP_Help/tle_def.html`, Sep 2011.

[33] eoPortal Satellite Missions Database. 2015.
Available at: `https://directory.eoportal.org/web/eoportal/satellite-missions`

[34] Andersen, D.E.: Computing NORAD Mean Orbital Elements From A State Vector. Tech. Rep., DTIC Document, 1994.

[35] STK High-Precision Orbit Propagator (HPOP). 2015.
Available at: `http://help.agi.com/stk/10.1.3/index.html?page=source%2Fhpop%2Fhpop.htm`

[36] Jochim, E., Gill, E., Montenbruck, O. and Kirschner, M.: GPS Based Onboard and Onground Orbit Operations for Small Satellites. *Acta Astronautica*, vol. 39, no. 9, pp. 917–922, 1996.

[37] Vallado, D.A. and Crawford, P.: SGP4 Orbit Determination. In: *Proceedings of AIAA/AAS Astrodynamics Specialist Conference and Exhibit*, pp. 18–21. 2008.

[38] Yunck, T.P., Wu, S.-C., Wu, J.-T. and Thornton, C.L.: Precise Tracking of Remote Sensing Satellites with the Global Positioning System. *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 28, no. 1, pp. 108–116, 1990.

[39] Tapley, B.D., Ries, J., Davis, G., Eanes, R., Schutz, B., Shum, C., Watkins, M., Marshall, J., Nerem, R., Putney, B. *et al.*: Precision Orbit Determination for TOPEX/POSEIDON. *Journal of Geophysical Research: Oceans (1978–2012)*, vol. 99, no. C12, pp. 24383–24404, 1994.

[40] TOPEX/Poseidon. Sep 2015.
Available at: `http://science.nasa.gov/missions/topex-poseidon/`

[41] Luthcke, S., Zelensky, N., Rowlands, D., Lemoine, F. and Williams, T.: The 1-Centimeter Orbit: Jason-1 Precision Orbit Determination Using GPS, SLR, DORIS, and Altimeter Data Special Issue: Jason-1 Calibration/Validation. *Marine Geodesy*, vol. 26, no. 3-4, pp. 399–421, 2003.

[42] Kuang, D., Bar-Sever, Y., Bertiger, W., Desai, S., Haines, B., Iijima, B., Kruizinga, G., Meehan, T. and Romans, L.: Precise Orbit Determination for CHAMP using GPS Data from BlackJack Receiver. In: *2001 ION National Technical Meeting Proceedings*. Session E1: Scientific Applications, Timing, and Frequency Long Beach, California, 2001.

[43] Van Helleputte, T. and Visser, P.: GPS based Orbit Determination using Accelerometer Data. *Aerospace Science and Technology*, vol. 12, no. 6, pp. 478–484, 2008.

[44] Kang, Z., Tapley, B., Bettadpur, S., Ries, J. and Nagel, P.: Precise Orbit Determination for GRACE using Accelerometer Data. *Advances in Space Research*, vol. 38, no. 9, pp. 2131–2136, 2006.

[45] Reigber, C., Schwintzer, P., Neumayer, K.-H., Barthelmes, F., König, R., Förste, C., Balmino, G., Biancale, R., Lemoine, J.-M., Loyer, S. *et al.*: The CHAMP-only Earth Gravity Field Model EIGEN-2. *Advances in Space Research*, vol. 31, no. 8, pp. 1883–1888, 2003.

[46] König, R., Reigber, C., Neumayer, K., Schmidt, R., Zhu, S., Baustert, G., Flechtner, F. and Meixner, H.: Satellite Dynamics of the CHAMP and GRACE LEOs as Revealed from Space- and Ground-based Tracking. *Advances in Space Research*, vol. 31, no. 8, pp. 1869–1874, 2003.

[47] Bock, H., Jäggi, A., Dach, R., Schaer, S. and Beutler, G.: GPS Single-Frequency Orbit Determination for Low Earth Orbiting Satellites. *Advances in Space Research*, vol. 43, no. 5, pp. 783–791, 2009.

[48] Choi, E.-J., Yoon, J.-C., Lee, B.-S., Park, S.-Y. and Choi, K.-H.: Onboard Orbit Determination using GPS Observations Based on the Unscented Kalman Filter. *Advances in Space Research*, vol. 46, no. 11, pp. 1440–1450, 2010.

[49] Mander, A. and Bisnath, S.: GPS-based Precise Orbit Determination of Low Earth Orbiters with Limited Resources. *GPS solutions*, vol. 17, no. 4, pp. 587–594, 2013.

[50] Reigber, C., Schwintzer, P., Stubenvoll, R., Schmidt, R., Flechtner, F., Meyer, U., König, R., Neumayer, H., Förste, C., Barthelmes, F. *et al.*: *A High Resolution Global Gravity Field Model Combining CHAMP and GRACE Satellite Mission and Surface Data: EIGEN-CG01C*. Geoforschungszentrum, 2006.

[51] van Daalen, D.C., jones, P.T. and Jaquet, M.C.: Advanced Automation 813: Course Notes, 2014. Stellenbosch University Postgraduate Course Notes.

[52] Thrun, S., Burgard, W. and Fox, D.: *Probabilistic Robotics*. MIT press, 2005.

[53] Ribeiro, M.I.: Kalman and Extended Kalman Filters: Concept, Derivation and Properties. *Institute for Systems and Robotics*, vol. 43, 2004.

[54] Labbe, R.R.J.: *Kalman and Bayesian Filters in Python*. Aug 2015.
Available at: `https://github.com/rlabbe/Kalman-and-Bayesian-Filters-in-Python`

[55] Runge-Kutta Method for Solving Two Coupled 1st Order Differential Equations or One 2nd Order Differential Equation. `http://www.phy.davidson.edu/FacHome/dmb/py200/RungeKuttaMethod.htm`, Oct 2014.

[56] Hoffman, J.D. and Frankel, S.: *Numerical Methods for Engineers and Scientists, Second Edition*. CRC Press, 2001.

[57] Franklin, G.F., Powell, J.D. and Workman, M.L.: *Digital Control of Dynamic Systems*. Addison Wesley Longman, Inc., 1998.

[58] IERS Bulletins. 2015.
Available at: `http://www.iers.org/IERS/EN/Publications/Bulletins/bulletins.html`

[59] Alburger, T.: The Origin of Julian Days. `http://www.magma.ca/~scarlisl/DRACO/julian_d.html`, Apr 2015.

[60] Meeus, J.: *Astronomical Algorithms*. 1st edn. Richmond, VA: Willmann Bell, 1991.

[61] Hashida, Y.: *Analytical Solution for Autonomous Determination of Near Circular Orbits*. PhD, University of Surrey, Apr 2003.

[62] Grewal, M.S., Weill, L.R. and Andrews, A.P.: *Global Positioning Systems, Inertial Navigation, and Integration*. 1st edn. John Wiley & Sons, 2001.

[63] The Future of Leap Seconds. `http://www.timeanddate.com/time/leap-seconds-future.html`, Apr 2015.

[64] Inertial Reference Frames and Coordinate Systems. `http://galileo.ftecs.com/stone-diss/chap3/inertial.html`, Oct 2002.

[65] Approximate Sidereal Time. `http://aa.usno.navy.mil/faq/docs/GAST.php`, Jun 2011. Astronomical Applications Department, U.S. Naval Observatory.

[66] du Plessis, J.: *Low Earth Orbit Satellite Propagators*. MEng, University of Stellenbosch, Dec 1999.

[67] Seidelmann, P.K.: *Explanatory Supplement to the Astronomical Almanac*. University Science Books, 1992.

[68] *Supplement to Department of Defense World Geodetic System 1984 Technical Report (TR 8350.2-A): Part 1 - Methods, Techniques, and Data Used in WGS 84 Development*, Dec 1987.

[69] Guard, U.C.: NAVSTAR GPS User Equipment Introduction. Sep 1996.

[70] Kaplan, E.D. and Hegarty, C.J.: *Understanding GPS: Principles and Applications*. Artech House inc., 2006.

[71] Remondi, B.W.: GPS-Toolbox. `http://www.ngs.noaa.gov/gps-toolbox/bc_velo/bc_velo.c`, Feb 2004.

[72] Langley, R.B.: Dilution of Precision. *GPS world*, vol. 10, no. 5, pp. 52–59, 1999.

[73] Liu, X., Jiao, W., Bai, Y. and Fan, J.: Global Coverage Performance Analysis Based on 4 BeiDou MEO Satellites. In: *China satellite navigation conference (CSNC) 2013 proceedings*, pp. 319–323. 2013.

[74] *SSTL SGR-05U Datasheet*, Sep 2015.
Available at: `http://www.sstl.co.uk/getattachment/97ae8ccc-024d-4376-a99d-7d3c2266a7f7/SGR-05U-05P`

[75] *DLR Phoenix-S Datasheet*, Sep 2015.
Available at: `http://www.dlr.de/rb/en/Portaldata/38/Resources/dokumente/GSOC_dokumente/RB-RFT/Phoenix_DataSheet_v11.pdf`

[76] *Spacequest GPS-12-V1 Datasheet*, Sep 2015.
Available at: `http://spacequest.com/products/GPS-12-V1.pdf`

[77] *Septentrio AsteRx-m Datasheet*, Sep 2015.
Available at: `http://www.septentrio.com/products/gnss-receivers/rover-base-receivers/oem-receiver-boards/asterx-m-oem`

[78] *Novatel OEM615 Datasheet*, Sep 2015.
Available at: `http://www.novatel.com/assets/Documents/Papers/OEM615.pdf`

[79] *Trimble COPERNICUS II Datasheet*, Sep 2015.
Available at: `http://www.mpi.ch/files/File/Trimble/Copernicus-II_DS_0508.pdf`

[80] *Fastrax Datasheets*, Sep 2015.
Available at: `http://www.glynstore.com/content/docs/fastrax/Fastrax%20Product%20Leaflet%20-%20September%202010.pdf`

[81] General Formula: Matrix Inversion in Block form. `http://www.cs.nthu.edu.tw/~jang/book/addenda/matinv/matinv/`, Mar 2001.

[82] Aggarwal, J.: *Digital Signal Processing*. Western Periodicals Company, 1979.

[83] Wickert, D.M.: *Introduction to Signals and Systems*. CRC Press, 2011.